This is to certify that the

thesis entitled

DEVELOPMENT OF SOFTWARE PACKAGE TO PREDICT TEMPERATURE VARIATION IN TRAILERS, BOXCARS, OR CONTAINERS

presented by

Raymond Lai-ming Cheung

has been accepted towards fulfillment of the requirements for

Master degree in Packaging

S. Paul Singh
Major professor

Date November 23, 1992

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE

MSU Is An Affirmative Action/Equal Opportunity Institution

c:\circ\datedue.pm3-p.1

DEVELOPMENT OF SOFTWARE PACKAGE TO PREDICT TEMPERATURE VARIATION IN TRAILERS, BOXCARS, OR CONTAINERS

By

Raymond Lai-ming Cheung

A Thesis

Submitted to

Michigan State University

in partial fulfillment of the requirements

for the degree of

Master of Science

School of Packaging

1992

ABSTRACT

DEVELOPMENT OF SOFTWARE PACKAGE TO PREDICT
TEMPERATURE VARIATION IN TRAILERS, BOXCARS, OR CONTAINERS

by

Raymond Lai-ming Cheung

Packaging engineers need to be well informed about the temperature levels and changes that a product would observe during transportation. This information is critical to design effective packaging and to ensure the safe arrival of products at destination.

The purpose of this thesis was to develop a user-friendly software for packaging engineers that predicts temperature changes for packaged products shipped in trailers, box cars, or containers. The program is written in 'C' computer language for the Windows™ environment. It provides the user with the familiar Macintosh like features such as dialog boxes for data entry and mouse for pointing device. The program uses weather data, geographic locations, product information, trailer capacity, travelling speed, etc., to predict the temperature variation of the packaged products while in transit.

ACKNOWLEDGEMENTS

I would like to thank Dr. S. Paul Singh, Dr. Gary Burgess, and Dr. Galen Brown for their guidance and advice. Especially Dr. Singh for his patience throughout the years.

I would also like to thank Dr. Charles Webb and Keith Williams for hiring me to work in the MSU Alumni Association as Automated Office Systems Assistant, and the rest of the alumni staff for their friendships and care. Without this employment opportunity, my last two years as a graduate student may not have become reality.

Finally, I would like to thank my parents, my brother, and my sister for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The objective of this research was to develop a user-friendly software application program (PACKTEMP) that uses temperature data from the Weather Bureau(UPS, 1976) to predict air temperature variation in a trailer. It was determined that Windows™ 3.0™ would be used as a desirable interface. The reasons are listed as below:

1.1 Advantage of Windows™ For the User:

IBM-DOS®/MS-DOS(Disk Operating System)™ has always been a hurdle for computer-illiterates because of the lengthy text-based commands and details it demands for a user to be efficient(Petzold, 1991). Windows™, however, revolutionizes the MS-DOS® computing environment by providing a Graphical User Interface(GUI) very similar to that of an Apple© Macintosh™ that has long been recognized for its user-friendliness. This new environment removes the barrier that has always separated the technology from the general user.

Windows™ does not replace DOS®. Windows™ builds itself above DOS® and makes DOS® transparent to the user. DOS® still does the file management, but Windows™ takes over the management of other resources such as communicating with the keyboard, monitor, memory, mouse, printer, etc. The user no longer is required to know specific commands to open a file, or start an application. Instead, the user simply points the cursor with the mouse to a certain icon, or selects a command from the menus offered by Windows™. The user-friendliness of Windows™ explains why it has emerged quickly as the most popular graphical user interface environment for MS-DOS® since its introduction in 1985.

Multitasking is another advantage Windows™ offers. As a user gets sophisticated, he/she would soon realize the need for fast switching between different software to accomplish the task at hand. For example, an accountant might transfer blocks of data from a spreadsheet to a word-processing document for a financial report; or a programmer might need to switch back and forth between a word processor and a compiler to debug and run a program. Under Windows™, every application has its own window, and can run concurrently with other programs. Not only can these programs coexist, Windows™ also allows easy data exchange between different applications. Text and graphics could be moved easily by selecting them with the mouse, and copied from one window to another, as long as the receiving application is able to handle the information.

Windows™ also provides a consistent fundamental look and feel for most applications. A rectangular box, scroll bar on the side, caption bar on top, dialog boxes, push buttons, etc. are used as shown in Figure 1. This kind of similarity makes the learning of a new software relatively easy. Once a Windows™ user learns the basics and meaning of the Windows™ commands, he/she can easily apply that knowledge to other programs.

Finally, the GUI provides a better use of the graphics capability of the computer. Windows™ graphics usage departs conceptually from the traditional use of the monitor as a device to echo keystrokes typed through the keyboard. WYSIWYG(what you see is what you get) video display of information shows the user exactly what the output would be like. Windows™ thus builds a much more intimate relationship between the user and a program by letting the user interact directly with the objects on the display.

1.2 Advantage of Windows™ For the Programmer:

The more user-friendly an environment is, the more details the programmer needs to attend to in general. In order to make the program flow easily, the programmer has to consider all the possible options a user might desire and code them into the program. Fortunately, Windows™, together with the Windows™ Software Development Kit(SDK)™, provides abundant built-in routines and library functions for the programmer to call upon. However, most programmers would initially find themselves lost, and then go through a mental reorientation, when they first start Windows™ programming. They may find the

programming complex at first, but once they become familiar with the new environment, they can easily adapt to the new functions and routines offered by this software.

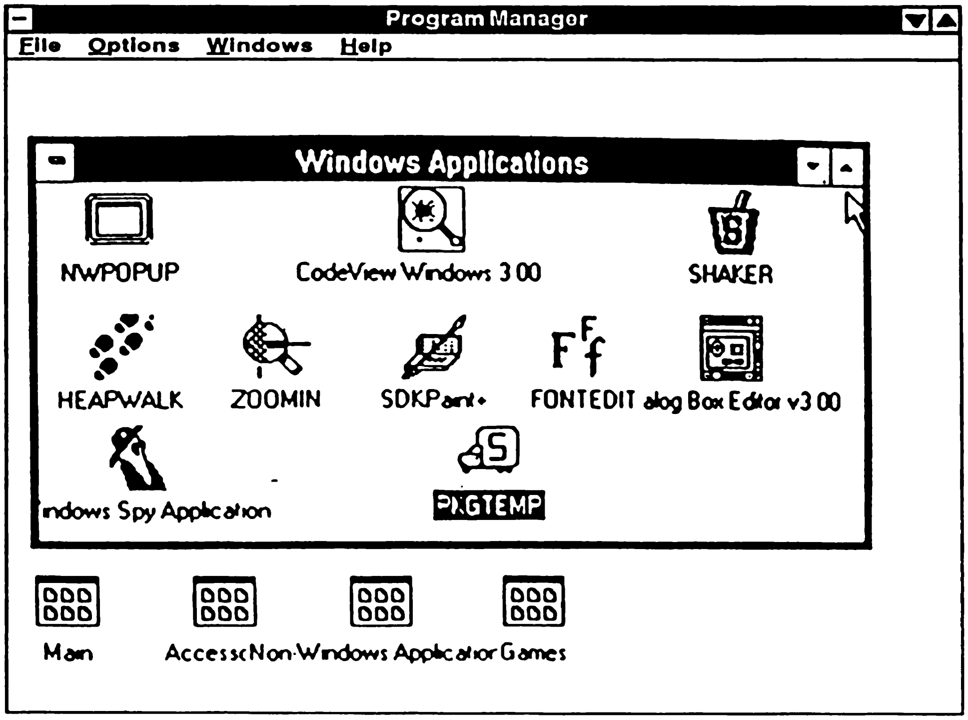


Figure 1. Windows Environment

2.0 Development of Software Package

The three basic phases of the development of a software package are definition, development, and maintenance(Figure 2). During the creation process of this program, which is named PACKTEMP, the following definitions and guidelines are followed.

2.1 Definition

During the definition phase of development, the question on **WHAT** the program is to accomplish needs to be answered. **WHAT** information needs to be processed; **WHAT** performance is desired; **WHAT** kind of design limitations exist; and **WHAT** user interface is efficient. These become the minimum requirements for the programmer to accomplish. The following are the definitions of PACKTEMP:

2.1.1 Audience of the program: PACKTEMP is particularly designed for personnel in the packaging and transportation industry who do not necessarily possess the computer skills to use the mathematical model developed by Burgess(1992) to predict temperatures in trailers.

2.1.2 Interface: PACKTEMP utilizes the user-friendly features in Windows™ such as pop-up menus, mouse, dialog boxes, etc. to minimize the complexity of

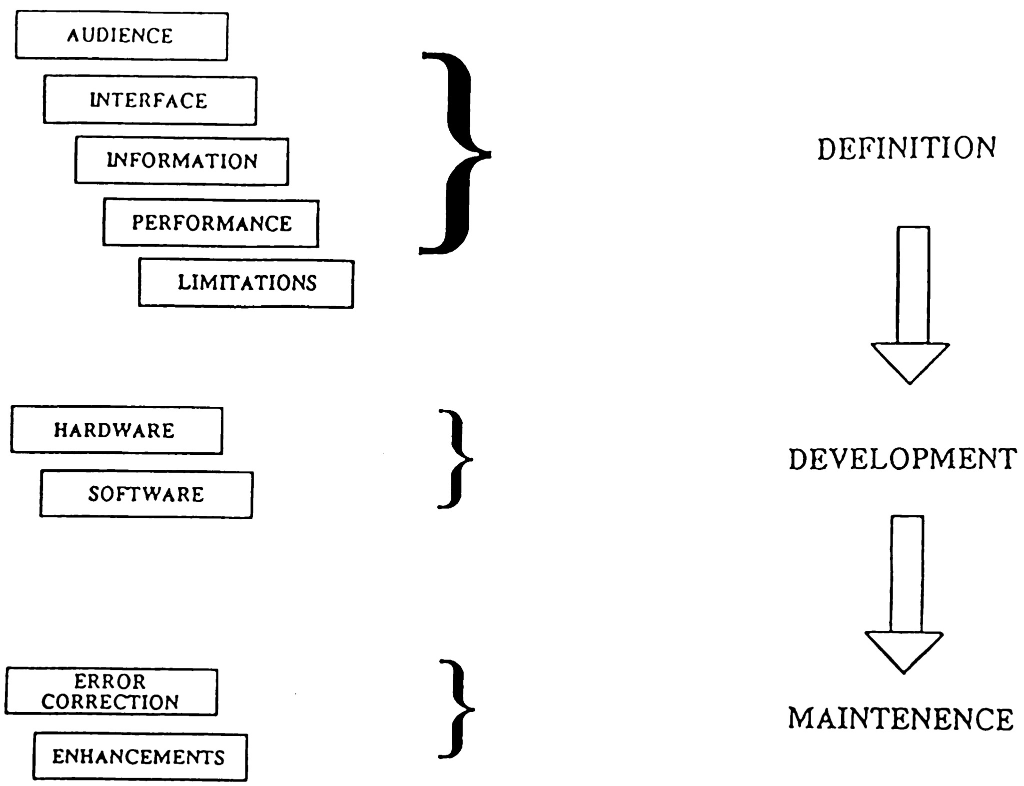


Figure 2. Flow chart of Program Development

operation demanded on the user as a program in BASIC would require. Multiple lines of data input can now be done within one single dialog box, and the flow of the program is completely menu driven. PACKTEMP also displays a map of the continental United States in order to provide the user with a visual image of the whereabouts of different cities at a glance.

2.1.3 Information to be processed: The following pieces of data need to be entered before the program can calculate the temperature variations.

i) Trailer: Size dimensions, cooling capacity, travelling speed, thermostat setting, and thermal conductivity of the trailer's wall material.

ii) Product: type, specific heat at given temperature, loading temperature.

iii) Weather information: Originating city, destination city, time of year, cloud coverage, relative humidity.

The program has a set of default values that were built into the code. These values can be easily changed by the user as more data is available for a particular shipment.

2.1.4 Performance: Due to the amount of computations involved with this simulation program, a IBM-PC compatible 386DX based machine is chosen as the platform. A math co-processor is recommended but not necessary. The output of the program provides the user the temperature variations that the product will observe for a given trip.

2.2 Development

At the development stage, the programmer has to determine how the guidelines and requirements set forth in this stage could be accomplished. The first step is to decide what platform to build the program on, and what software and programming language to use. PACKTEMP was written using the following hardware and software, after a careful selection process:

Hardware:

i) CompuAdd 325 - 100% IBM™ compatible personal computer with a 386 processor at 25 MHz

ii) 40 megabyte hard disk

iii) 4MB RAM

iv) VGA color monitor

Software:

i) Microsoft Windows™ 3.0

ii) Software Development Kit(SDK) 2.0

iii) Microsoft C Compiler Version 6.0

Windows™ was chosen to be the preferred environment because of its user-friendly features. SDK is an additional package that provides a Windows™ application programmer with built-in functions and library routines. The 'C' computer language was used as the programming language because of its compatibility with both the Windows™ programming environment and SDK.

2.3 Maintenance

This final phase of the project focuses on the 'change' in the software that is needed to keep up with the ever-evolving computing environment. Change is also needed when the end-user demands additional features and enhancements in the software. Finally, no matter how much testing the programmers carry out, it is very likely that the end user will uncover bugs that were overlooked. Since at the time this thesis is being prepared, there are no reports of defects or need of change in PACKTEMP, the maintenance phase has therefore not yet begun. However, recommendations for improvement were suggested by faculty members at the thesis defense. These suggestions would be discussed in greater detail in Chapter 4.

2.4 Development of PACKTEMP

2.4.1 Drawing of the U.S. map on screen:

i) The coordinates of the U.S. state lines were converted and scaled to screen coordinates.

ii) The screen coordinates are used as the arguments to the 'LineTo()' and 'MoveTo()' functions provided by SDK.

iii) The coordinates of the cities of interest are converted also into screen coordinates. The airport code names of these cities are then written on the screen using the SDK function 'TextOut()'.

2.4.2 Building the databases

Two major arrays were built to store all the vital information for the program, namely 'PRODARR' and 'CTEMPARR'. 'PRODARR' stores the information of the product being transported. 'Ratea' and 'Rateb' are the two coefficients used by the program to extrapolate the real time respiration heat at various temperatures. 'Heatc' is the specific heat capacity. 'CTEMPARR' stores the name of the cities, the actual coordinates, screen coordinates, city name abbreviations, and the temperature coefficients. The use of these pre-built data arrays minimizes data input, thus enabling a friendly interaction between the user and the program.

2.4.3 Designing the pulldown menu and dialog boxes

The menus and dialog boxes are means of getting around and entering data in PACKTEMP. They will be discussed further in this chapter.

2.4.4 The CDP program in BASIC was translated into 'C' and tested independently before being incorporated into the Windows™ procedures. Appendix B lists the actual 'C' code of the whole PACKTEMP program.

2.5 Description of Input

The inputs to PACKTEMP look somewhat different than a program in BASIC would. The most significant improvement is that it allows a user to re-enter a piece of information without starting over from the beginning. This helps to alleviate frustration by a user who is unfamiliar with the keyboard and may commit typographical errors.

When a user clicks on the menu bar (Figure 3) for the pulldown menu that initiates the dialog box for data input, the dialog box appears with all the information that needs to be addressed by the user. Each question comes with a default answer that was preselected in the program. A user can then use the mouse or keyboard to edit the default data to input the specific data desired for the test run. At any point in time, the user can point the mouse back to any question and change the input. When all the questions have been answered,

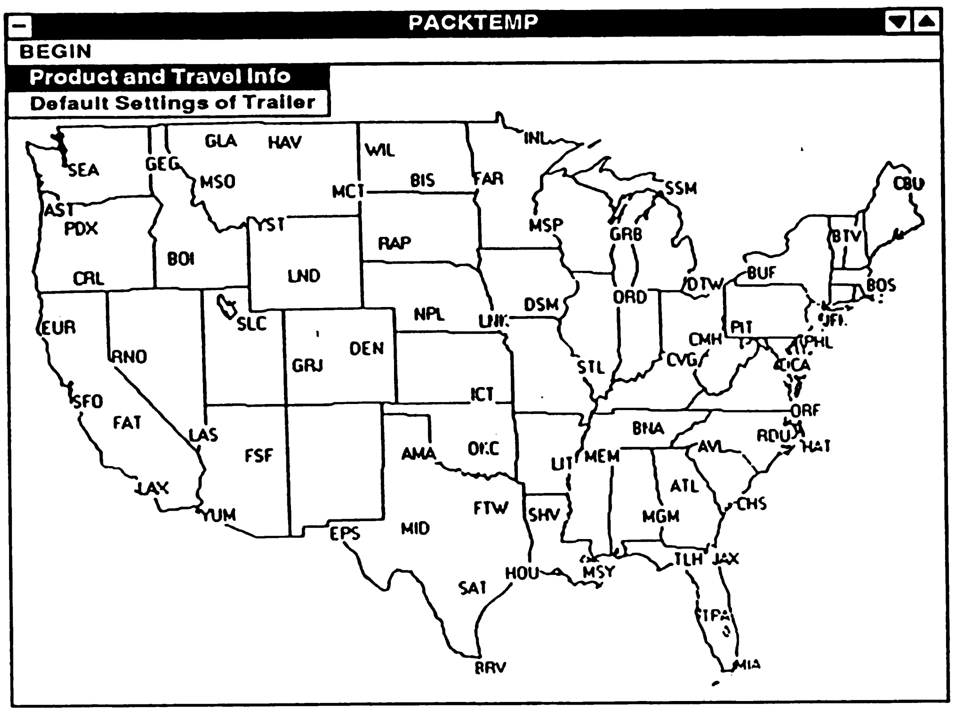


Figure 3. Menu Bar

the user can then click one of the three buttons at the bottom of the screen. 'OK' saves the data and returns to the 'MAP' screen; 'RECALC' saves the data, but also performs the calculations with the specified data and displays the results on screen; 'CANCEL' simply disregards the changes made to thedefault data and return to the 'MAP' screen.

The input data are grouped together in two major categories. 'PRODUCT & TRAVEL INFO' dialog box (Figure 4) contains all relevant information about the start and destination cities, time of year, product type, product specific heat capacities, etc. 'DEFAULT SETTINGS OF TRAILER' dialog box (Figure 5) prompts the user for the trailer dimensions, speed of travel, etc.

The following are the questions asked in the two dialog boxes and a description of the input expected for the program. All numeric input values to be used are integers. Floating points or character values may cause unpredictable results and possible loss of data.

2.5.1 PRODUCT & TRAVEL INFO Dialog Box:

Q1: Start City: Use Table 1 to determine the city number of all the 73 cities available for this program. Start city refers to the city where the trailer is originating from.

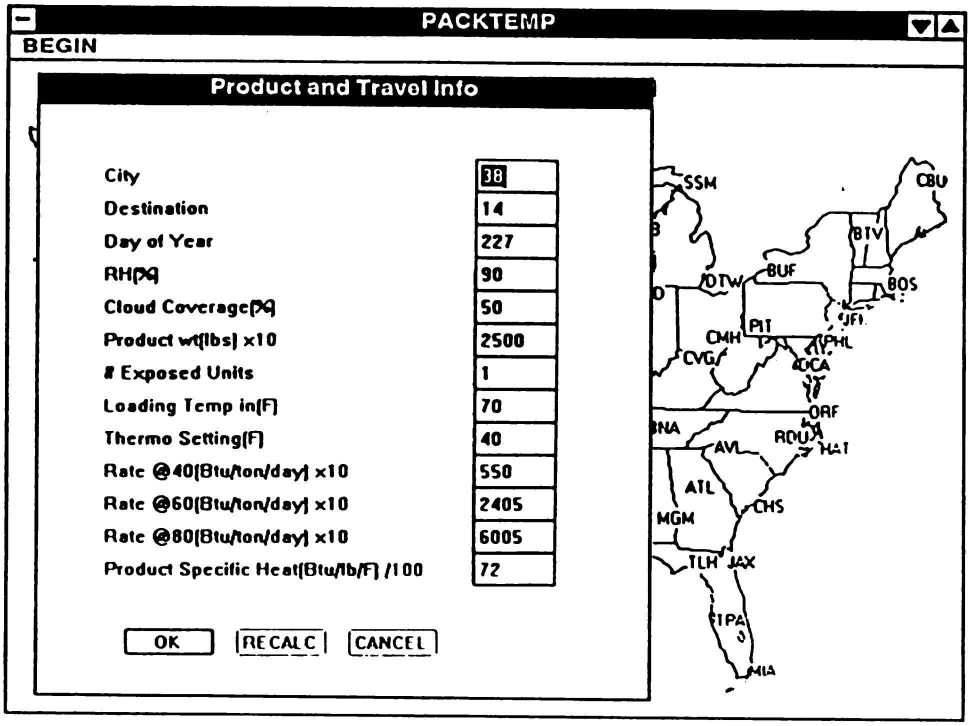


Figure 4. Product and Travel Information Dialog Box

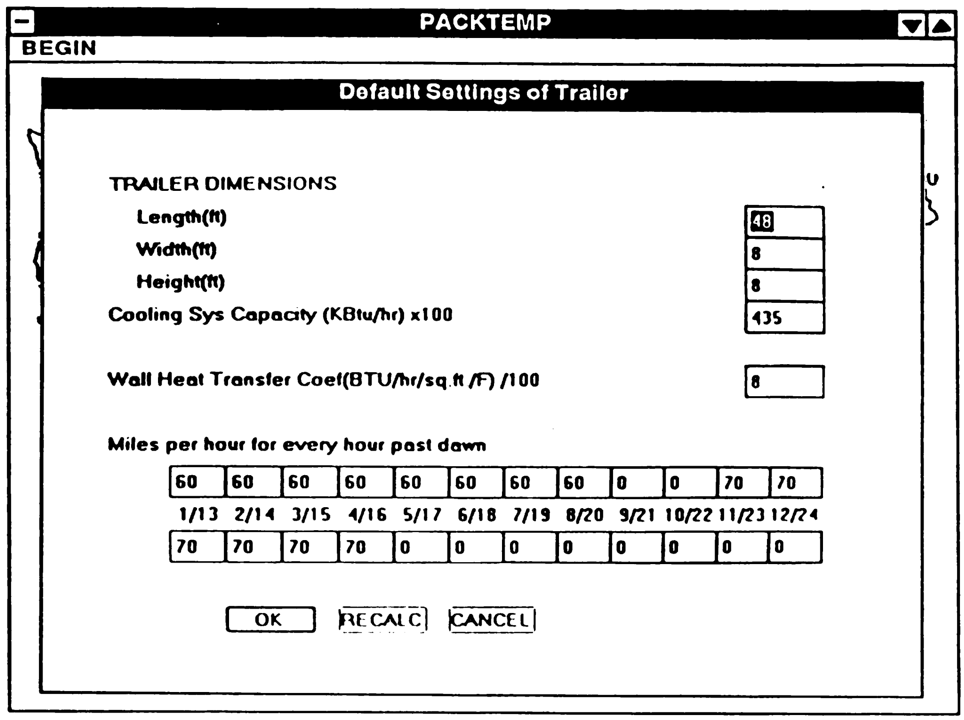


Figure 5. Default Settings of Trailer Dialog Box

Q2: Destination City: Enter the number describing the destination city(Table 1). The originating city, together with the destination city provides the program with the orientation of the trailer.

Q3: Day of year: Use Table 2 to determine what day of the year the shipment begins. For example, if the trailer departs on February 10, day of year would be 31(number of days in January) + 10(ten days in February) = 41. It should be noted that the average high and low temperatures of the month of shipment are used in the program to as outside temperatures.

Q4: Relative Humidity(%): The relative humidity entered here is used to modify the BTU rating of the refrigeration unit used. The higher the RH is, the less efficient is the refrigerator because more energy is used to condense water vapor. This RH can refer either to the outside air RH, or the RH inside the trailer. Use whichever is greater. If there is no refrigeration used, the RH data would be ignored in the program.

Q5: Cloud Coverage(%): The program uses this information to account for the diminished effect of solar radiation on the trailer surfaces depending

Table 1: List of city names,numbers,and initials used in PACKTEMP

CITY NUMBER	INITIALS	CITY NAME
1	AMA	AMARILLO
2	AVL	ASHVILLE
3	AST	ASTORIA
4	ATL	ATLANTA
5	BIS	BISMARK
6	BOI	BOISE
7	BOS	BOSTON
8	BRV	BROWNSVILLE
9	BUF	BUFFALO
10	BTV	BURLINGTON
11	CBU	CARIBOU
12	CHS	CHARLESTON
13	ORD	CHICAGO
14	CVG	CINCINNATI
15	CMH	COLUMBUS
16	CRL	CRATER LAKE
17	DEN	DENVER
18	DSM	DES MOINES
19	DTW	DETROIT
20	EPS	EL PASO
21	EUR	EUREKA
22	FAR	FARGO
23	FSF	FLAGSTAFF
24	FTW	FORT WORTH
25	FAT	FRESNO
26	GLA	GLACIER
27	GRJ	GRAND JUNCTION
28	GRB	GREEN BAY
29	HAT	HATTERAS
30	HAV	HAVRE
31	HOU	HOUSTON
32	INL	INTERNATIONAL FALLS
33	JAX	JACKSONVILLE
34	LND	LANDER
35	LAS	LAS VEGAS
36	LNK	LINCOLN

Table 1. (cont'd)

CITY NUMBER	INITIAL	CITY NAME
37	LIT	LITTLE ROCK
38	LAX	LOS ANGELES
39	MEM	MEMPHIS
40	MIA	MIAMI
41	MID	MIDLAND
42	MCT	MILES CITY
43	MSP	MINNEAPOLIS
44	MSO	MISSOULA
45	MGM	MONTGOMERY
46	BNA	NASHVILLE
47	MSY	NEW ORLEANS
48	JFK	NEW YORK
49	ORF	NORFOLK
50	NPL	NORTH PLATTE
51	OKC	OKLAHOMA CITY
52	PHL	PHILADELPHIA
53	PIT	PITTSBURGH
54	PDX	PORTLAND
55	RDU	RALEIGH
56	RAP	RAPID CITY
57	RNO	RENO
58	SLC	SALT LAKE CITY
59	SAT	SAN ANTONIO
60	SFO	SAN FRANCISCO
61	SSM	SAULT STE MARIE
62	SEA	SEATTLE
63	SHV	SHREVEPORT
64	GEG	SPOKANE
65	STL	ST LOUIS
66	TLH	TALLAHASSEE
67	TPA	TAMPA
68	DCA	WASHINGTON
69	ICT	WICHITA
70	WIL	WILLISTON
71	YST	YELLOW STONE
72	YUM	YUMA

Table 2: Day of Year Determination

January 1	1
Feburary 1	32
March 1	60
April 1	91
May 1	121
June 1	152
July 1	182
August 1	213
September 1	244
October 1	274
November 1	305
December 1	335

e.g. To find out the day of year for June 20:
152 (Jun 1) + 19 = 172

on the cloud coverage. Average of 100% signifies a completely overcast day, and 0% a clear day. The program, however, would still assign a minimum of 10% in case of complete overcast because of diffuse solar radiation.

NOTE: Enter 50 for 50% coverage.

Q6: Product Weight(lbs): This refers to the net weight of the product, not the gross weight. This value, together with loading temperature and specific heat of the product, determines the field heat in the load as well as the ability to retain or lose heat.
NOTE: Enter this value as a factor of 10.
i.e. If the load is 25000 lbs, enter 2500.

Q7: Exposed units: The number of exposed units bears a direct relationship with the rate of heat transfer: the more units exposed means the more surface area of the load is available for heat transfer. A minimum of 3 inches between parts of load is needed to label these parts as separate units because tightly packed configurations do not allow air to freely circulate between individual pallet loads.

Q8: Loading temperature(F): This is the temperature at which the product is loaded into the trailer.

Loading temperature, together with product weight and specific heat, determine the field heat(existing BTU's) in the load. The program's calculation refers to this as a starting point.

Q9: Thermo Set(°F): This is the desired temperature for transportation of the product, and also the temperature setting on the thermostat of the refrigeration/heating unit. Table 4 lists the recommended transit temperatures for a variety of agricultural products. If the trailer does not have a temperature regulatory unit or if it is not being turned on for certain temperature insensitive products, the Thermo Set value will be ignored. Be sure, however, to enter 0 for BTU on question Q2 in dialog box for Default Settings of the Trailer.

Q10: Respiration Rate @ 40, 60, 80°F in BTU/ton/day: Due to the respiration process, fresh produce generates heat. The rate of respiration varies with the temperature of which they are stored. This heat that is produced by the product acts as a heat source inside the trailer and impacts the overall heat balance. Generally speaking, the higher the respiration, the more heat is produced. The PACKTEMP program uses the respiration rate of the product involved at the three temperatures to interpolate and extrapolate the rates at other temperatures.

See Table 3 for more information on respiration rate value.

Q11: Product Specific Heat in BTU/lb/°F: Specific heat(C_p) is defined as the amount of heat energy(BTU) required to raise the temperature of 1 pound of the material by 1°F. Products with high specific heat, such as water, require much more heat input to cause a temperature rise. On the same token, the refrigeration unit needs to remove much more heat, for a product that has a high specific heat to lower its temperature by 1°F. If a load consists of a variety of materials that have a different specific heat values, the weighted average of the specific heat should be used. Example: A load of 5000 lbs of wood(C_p = 0.67), 10,000 lbs of aluminum(C_p = 0.21), and 15,000 lbs of plastic(C_p = 0.5) would have the following specific heat value:

(0.67*5,000+0.21*10,000+0.50*15,000)/30,000 = 0.43

NOTE: Enter the specific heat value as a factor of 100. i.e. if the actual value is 0.43, enter 43.

Consult Table 4 and Table 5 for Specific Heat

Table 3. Approximate Amount of Respiration Heat Produced by Fruits and Vegetables at Various Temperatures

Commodity	BTU/ton/day x 10		
	40°F	60°F	80°F
Apples	135	490	-
Apricots	505	1170	-
Artichokes	1045	2640	5005
Asparagus	1805	3850	9325
Avocadoes	550	2405	6005
Bananas(green)	-	485	740
Beans - Green Snap - Lima(in pod)	 1030 610	 3810 2470	 - -
Beets	410	720	-
Blackberries	795	-	-
Blueberries	235	1055	2225
Broccoli, sprouting	2140	5650	15840
Brussels sprouts	770	2200	-
Cabbage	220	490	1235
Carrots	430	875	-
Cauliflowers(trimmed)	450	1010	2465
Celery	240	820	-
Cherries - Sour - Sweet	 285 260	 850 770	 1365 -
Corn(sweet)	1385	3585	7890
Cranberries	950	-	-
Cucumbers	-	530	805
Figs(fresh)	265	1235	2100
Gooseberries	285	595	-
Grapefruit	100	310	-
Grapes - American - European	 120 100	 350 240	 850 605
Kale(whole leaves)	890	3025	-
Leeks	535	2195	2485
Lemons	125	365	535
Lettuce - Head - Leaf	 365 645	 845 1380	 1810 3220
Limes	80	180	665

Table 3. (cont'd)

Commodity	BTU/ton/day x 10		
	40°F	60°F	80°F
Mangoes	350	990	2640
Melons - Cantaloupes - Honeydew - Watermelons	 205 90 80	 795 305 -	 1470 670 -
Mushrooms	1560	4600	-
Okra	1225	3205	7590
Onions - dry - green	 75 940	 240 1795	 620 3380
Oranges	120	400	715
Parsnips	290	825	-
Peaches/Nectarines	170	830	2235
Pears - Bartlett - Kieffer	 165 -	 825 385	 - 530
Peas(green in the pod)	1445	4190	7920
Peppers(sweet)	290	850	1210
Pineapples	40	345	1080
Plums(including fresh prunes)	145	270	1090
Potatoes - Uncured - Cured	 260 125	 485 195	 - -
Radishes(topped)	210	710	1640
Raspberries	765	2020	-
Rhubarb(w/o leaves)	320	870	-
Romaine	455	975	2385
Spinach	1015	3935	-
Squash - Butternut - Yellow Straightneck	 - 360	 - 1825	 - 4180
Strawberries	545	1795	4180
Sweet Potatoes - Cured - Uncured	 - -	 480 630	 - 1400
Tomatoes - Mature green - Pink	 145 130	 490 585	 940 905
Turnips	215	500	-
Watercress	1015	4070	-

Values of select food items. Consult Table 6 for non-perishable items.

2.5.2 DEFAULT SETTINGS OF TRAILER Dialog Box:

Q1: Trailer Dimensions: The surface area/size of the trailer determines the amount of heat gained/lost through the skin, or by convection to the outside/inside air. The dimension specified should be the inner dimension of the trailer.

Q2: Cooling System Capacity(BTU/hr): Most refrigeration units manufactured for domestic trailers have a rating of 43,500 BTU/hr. For a non-refrigerated trailer, enter zero. Enter BTU as a negative number if the trailer is heated.

NOTE: Enter the BTU value as a factor of 10.

i.e. If the actual value is 43500, enter 4350.

Q3: Wall heat transfer Coefficient(BTU/hr/ft^2/°F):

This value determines how well the trailer wall conducts heat. The higher the value, the more easily heat is gained or lost through the trailer wall. Use the thermal conductivity properties in Table 6 together with the following procedure to determine the coefficient of any type of trailer wall construction:

Example 1:

Trailer Wall: 1/16" of aluminum (0.0052 ft)

3.00" of foam (0.25 ft)

Sum of thickness-to-conductivity ratio=

.0052/118 + .25/.02 = 12.5

therefore, the wall heat transfer coefficient is $1/12.5 = 0.08$ BTU/hr/ft^2/°F

Example 2:

Trailer Wall: 1/32" aluminum outside (.0026 ft)

3/4" plywood (.0625 ft)

Sum of thickness-to-conductivity ratio=

.0026/118 + .0625/.13 = .48

therefore, the wall heat transfer coefficient is $1/.48 = 2.08$ BTU/hr/ft^2/°F

NOTE: Enter the coefficient at a factor of 100. i.e. enter 72 if the actual value is 0.72.

Q4: Miles Per Hour for Every Hour Past Dawn: The wall heat transfer coefficient of the trailer varies depending upon whether the trailer is stationary or moving. A moving trailer dissipates more heat when travelling in a hot summer day due to convection. PACKTEMP is designed to accept variations in trailer speed every hour past dawn. Every entry is the average speed of the trailer during that hour. Using zeroes tells PACKTEMP that the trailer is stationary at all times. PACKTEMP also points the trailer toward its destination. If the start and

Table 4. Specific Heat Above and Below Freezing of Select Items and the Recommended Transit Temperatures

Fruits/Vegetables	BTU/lb/°F		
	Above Freezing	Below Freezing	Recommended Transit Temperature
Apples	0.87	0.45	30-40
Apricots	0.88	0.46	32
Artichokes	0.87	0.45	32
Asparagus	0.94	0.48	32-35
Avocadoes	0.72	0.40	40-55
Bananas	0.80	0.42	56-58
Beans - Green Snap - Lima	 0.91 0.73	 0.47 0.40	 40-45 37-41
Beets	0.90	0.46	32
Blackberries	0.88	0.46	31-32
Blueberries	0.86	0.45	31-32
Broccoli	0.92	0.47	32
Brussels sprouts	0.88	0.46	-
Cabbage	0.94	0.47	32
Cantaloupes	0.93	0.48	36-41
Carrots	0.90	0.46	32
Casaba/Crenshaw Melons	0.94	0.48	-
Cauliflower	0.93	0.47	32
Celery	0.95	0.48	32
Cherries	0.87	0.45	30-32
Corn(sweet)	0.79	0.42	32
Cranberries	0.90	0.46	36-40
Cucumbers	0.97	0.49	50-55
Eggplant	0.94	0.48	46-54
Endive/escarole	0.94	0.48	32
Figs(fresh)	0.82	0.43	-
Garlic(dry)	0.69	0.40	32-34
Gooseberries	0.90	0.46	-
Grapefruits	0.90	0.46	50-60
Grapes	0.86	0.45	30-32
Honeydew melons	0.94	0.48	-

Table 4. (cont'd)

Fruits/Vegetables	BTU/lb/°F		
	Above Freezing	Below Freezing	Recommended Transit Temperature
Kale	0.89	0.46	32
Leeks	0.88	0.46	-
Lemons	0.91	0.46	50-55
Lettuce(head)	0.96	0.48	-
Limes	0.89	0.46	48-50
Mangoes	0.85	0.44	55
Mushrooms	0.93	0.47	32
Nectarines	0.86	0.44	31-32
Okra	0.92	0.46	45-50
Onions(dry)	0.90	0.46	32
Oranges	0.90	0.46	32-48
Parsley	0.88	0.45	32
Parsnips	0.84	0.44	32
Peaches	0.91	0.46	31-32
Pears	0.86	0.45	32
Peas	0.79	0.42	32
Peppers	0.94	0.47	45-55
Pineapples	0.88	0.45	45-55
Plums/fresh prunes	0.88	0.45	32
Potatoes - early-crop - late-crop	 0.85 0.82	 0.44 0.43	50-60
Pumpkins	0.92	0.47	50-55
Radishes	0.95	0.48	32
Raspberries - black/red	0.84/0.87	0.44/0.45	32
Rhubarb	0.95	0.48	32
Spinach	0.94	0.48	32
Squash - Summer/Winter	0.95/0.88	0.48/0.45	41-50/50-55
Strawberries	0.92	0.47	30-32
Sweet Potatoes	0.76	0.41	55-60
Tangerines	0.90	0.46	40
Tomatoes	0.95	0.48	46-50
Watermelons	0.97	0.48	50-60

Table 5. Specific Heat Above and Below Freezing of Select Dairy Products

Dairy, meat products	BTU/lb/F	
	Above freezing	Below freezing
Bacon	0.38	0.26
Beef(fresh)	0.77	0.41
Butter	0.36	0.25
Cheese	0.52	0.31
Eggs(shell)	0.73	0.40
Egg solids(whole)	0.22	0.21
Egg yolk solids	0.23	0.21
Ice cream	0.70	0.39
Hams: - Cured - Fresh	 0.67 0.53	 0.37 0.31
Lamb(fresh)	0.72	0.40
Milk	0.93	0.46
Oleomargarine	0.32	0.25
Pork(fresh)	0.53	0.31
Poultry(fresh)	0.80	0.42
Smoked sausage	0.62	0.35
Veal(fresh)	0.74	0.40

destination city are both the same, PACKTEMP would point the trailer north. This orientation determines how much solar radiation the trailer is exposed to due to the general direction of travel. In situations that the trailer is parked for an extended period of time, the user can input zero for speed for every hour to tell the program that the trailer is stationary.

When the 'RECALC' button is hit after all the data is entered, the program recalculates using the new information given and generates the results.

Table 6. Thermal Conductivities for Selected Materials

MATERIAL	CONDUCTIVITY(BTU/hr/ft/°F)
ALUMINUM	118.000
BRICK	0.60
CORK	0.025
FOAM	0.020
GLASS	0.450
RUBBER	0.090
STEEL	26.000
WOOD	0.130

3.0 Use of Software Package

This chapter describes how this software may be used by a user.

3.1 Installation

Before installing the software, be sure your equipment satisfies the following minimum requirements:

Compatibility:	100% IBM compatible
Microprocessor:	386sx or better
RAM:	640K or more
Software:	DOS 3.3 or later version
	Windows 3.0 or later version
Mouse:	Recommended
Display:	CVGA recommended

Follow these steps during installation:

1, Insert program disk in floppy drive.

2, Copy temp.exe onto your hard disk - either to the Windows subdirectory, or any other directory.

3, Start Windows.

4, Create a new program item in the Windows Applications folder by selecting NEW under FILE(Figure 6).

5, Enter a name for the program(PACKTEMP, or anything you like).

6, Enter complete path of where you placed the temp.exe file on the hard disk.

7, Double click the Trailer icon to start(Figure 7).

8, Follow the instructions in this thesis to run the program.

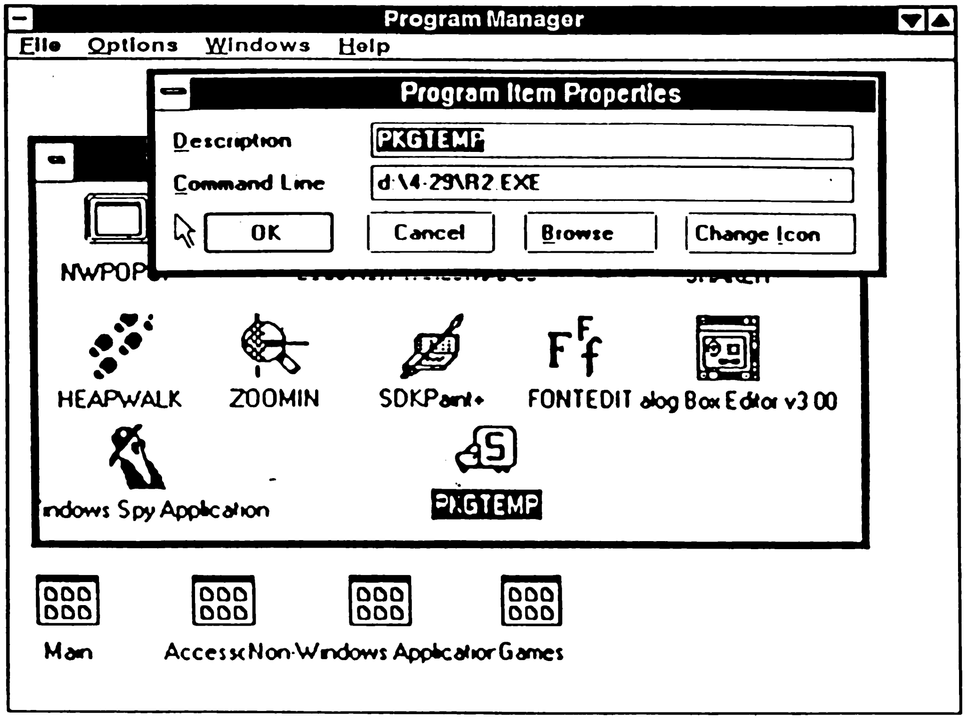


Figure 6. Dialog Box Showing Path to PACKTEMP

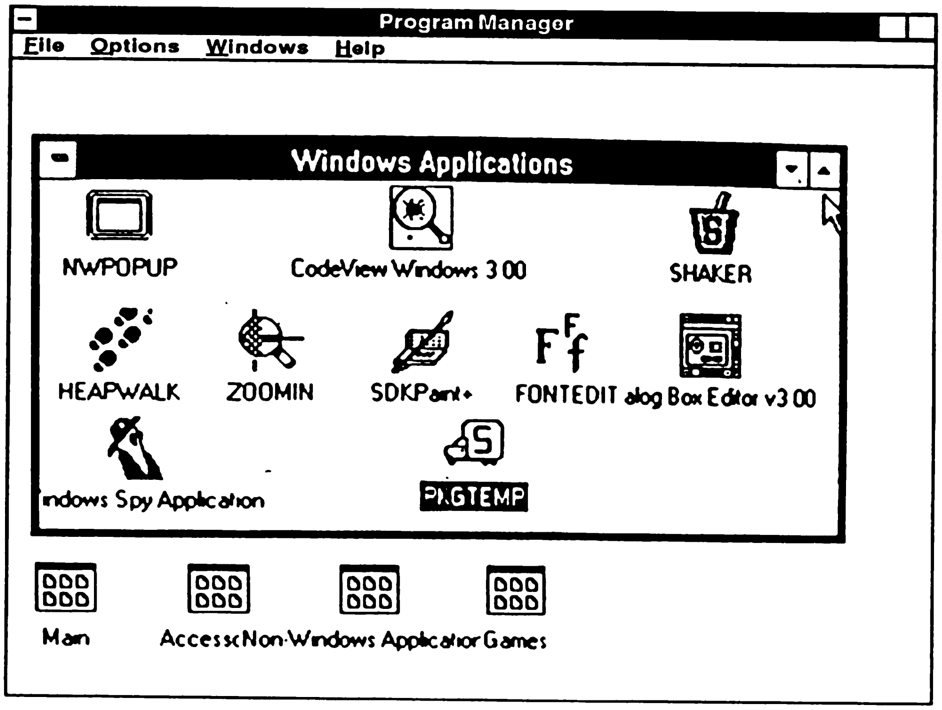


Figure 7. PACKTEMP Icon

3.2 General Instructions

Once the map of the United States is displayed on the screen(Figure 8), you will see one 'up' and one 'down' arrow at the top right hand corner of the window. The 'up' arrow would open up the current window to full screen, while the 'down' arrow would temporarily close the current window while keeping it active. The user is suggested to open the window into the full screen mode to display all the cities.

Use the mouse to point the cursor to BEGIN below the caption bar on the upper left corner and click. A menu would then appear showing choices of Product and Travel Info, Default Settings of Trailer, and Output as shown in Figure 9.

The first two choices, Product and Travel Info and Default Settings of Trailer, each contains a set of questions about the weather, product, etc. Using the descriptions and help from the previous chapter, input the values desired for each question. You can use the mouse to point the cursor to a particular box, or you can use tab to cycle the cursor through all input boxes. The values that appear in the boxes when the dialog box is first pulled up are default values that are set every time the program is first started. After changing and entering all the values deemed necessary for the particular run, you can click one of the three grey buttons at the bottom of the dialog box:

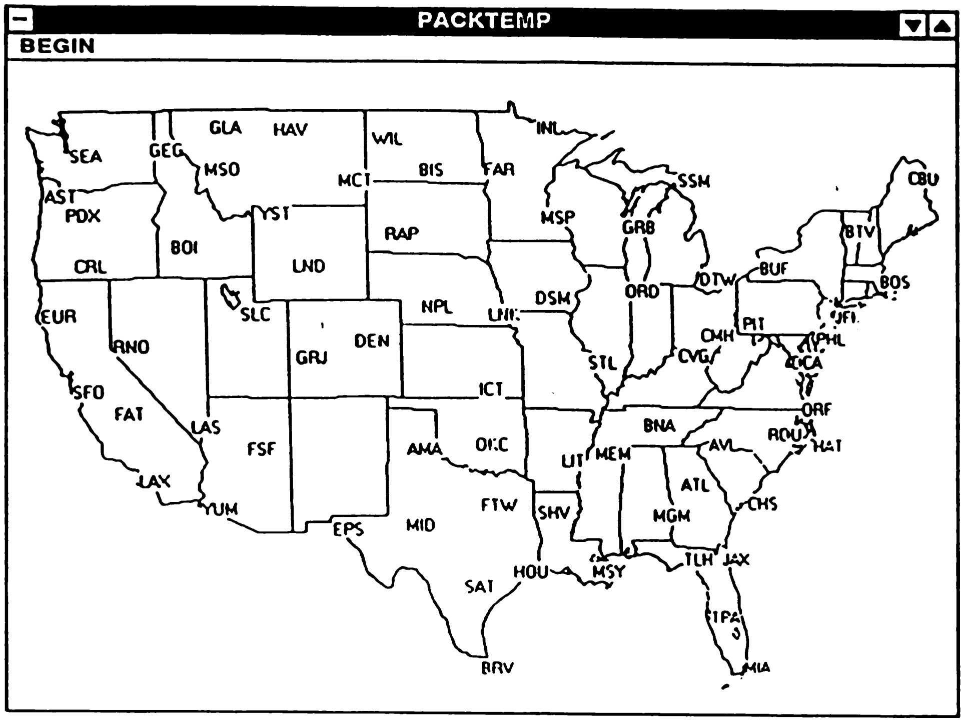


Figure 8. PACKTEMP Screen

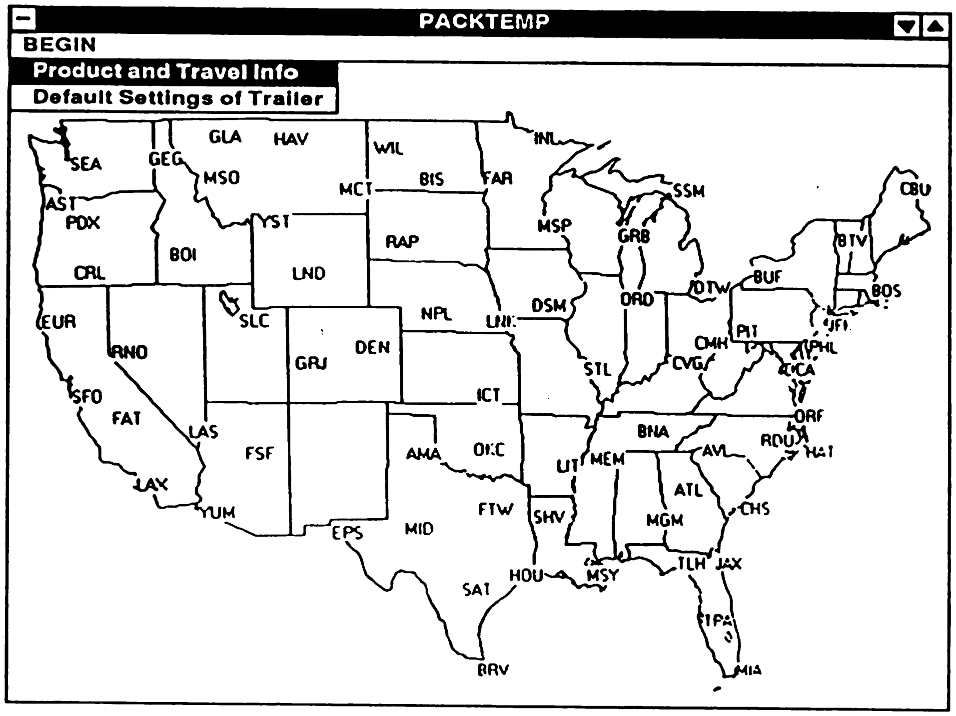


Figure 9. PACKTEMP Screen with Pulldown Menu

OK: OK informs the program that the values entered aregood and could be used for calculation. The program would save all values currently in the dialog box for use in later phase of the run. After clicking OK, the dialog box would disappear and the map would be brought back as the background. If in fact the user chooses to go back again to the same dialog box, the values previously entered would now reappear.

CANCEL: CANCEL would exit the dialog box without saving any of the changes made to the default values. The map would again be brought back as background.

RECALC: RECALC would first save the current values entered, then perform a calculation with these values. When done, the dialog box would disappear and a grey box with the output values would appear on top of the map showing the output values. The user can examine these values, then go back to the first two options on the menu to change certain parameters. When done viewing, the user can use the right button on the mouse to redraw the screen and get rid of the grey box.

To quit the PACKTEMP program, select CLOSE from the FILE menu, or hit ALT+F4.

4.0 CONCLUSION

This project successfully developed a computer application program 'PACKTEMP'. This program is capable of predicting air temperature variations in trailers when traveling between major cities in the United States. The program is based on the mathematical model developed by Burgess(1992) and is interactive through Windows™.

The faculty committee of this research study recommended the following as future enhancements of the program:

1. Program could read in the time of year in the form of calendar date (e.g. March 12), and then calculate the actual day of the year.
2. The heat transfer coefficients and thicknesses of materials used to construct the trailer wall be directly entered into the 'Default Settings of Trailer' dialog box, and have the program calculate the actual combined heat transfer coefficient.
3. Expand program for worldwide applications.

This program is currently being used by eight major corporations in the United States who are members of the Consortium of Distribution Packaging for a period of one year.

APPENDIX A

This section lists a number of tables demonstrating the effect of variables such as load temperature, number of exposed units, etc. in the outcome of a test run. The input data is identical to that listed in figures 2 and 3 in Chapter 2.

i, Table 7 - sample run.

ii, Table 8 - change in the number of exposed units.

iii, Table 9 - change in the loading temperature.

iv, Table 10 - the chiller is shut off by putting a zero in the BTU rating box.

v, Table 11 - the composition of the trailer wall material is altered. The foam is now replaced with plywood.

vi, Table 12 - in order to show the effects in the next 24 hours of the trip, the final load temperature of the previous day is used as the starting load temperature.

vii, Table 13 - the product is now 15,000 lbs of steel with no refrigeration, respiration, 0 cloud coverage, and stationary.

Table 7. Sample Run 1.

Distance = 2197.279 miles
Available Daylight = 13.3586 hrs
Expected min and max temperatures ate 64°F and 81°F

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	66.1	69.3	100.0
2	120.0	68.3	69.5	68.6	100.0
3	180.0	70.9	72.9	67.8	100.0
4	240.0	73.4	76.0	67.1	100.0
5	300.0	76.0	78.9	66.3	100.0
6	360.0	78.5	81.6	65.6	100.0
7	420.0	80.0	82.9	64.8	100.0
8	480.0	79.0	81.7	64.0	100.0
9	480.0	78.0	94.5	63.2	100.0
10	550.0	77.0	88.4	62.4	100.0
11	620.0	76.0	77.1	61.4	100.0
12	690.0	75.0	75.7	60.5	96.9
13	760.0	74.1	73.9	59.6	92.4
14	830.0	73.1	72.8	58.7	88.4
15	900.0	72.1	71.9	57.8	84.6
16	900.0	71.1	70.9	57.0	80.9
17	900.0	70.1	68.4	56.2	76.8
18	900.0	69.1	67.5	55.5	73.5
19	900.0	68.1	66.6	54.8	70.2
20	900.0	67.2	65.6	54.1	67.1
21	900.0	66.2	64.7	53.5	64.2
22	900.0	65.2	63.8	52.9	61.3
23	900.0	64.2	62.9	52.3	58.6
24	900.0	63.2	61.9	51.7	56.0

NOTE: 1, The day August 15 is 13 hrs, 21 minutes long.
2, The calculated distance between Los Angeles and Cincinnati is 2197 miles.
3, The average of the low temperatures is 64°F and high is 81°F.
4, The outside air temperature reaches its maximum at the sun's zenith(half the day length), and its minimum at dawn & 24 hours after dawn.
5, The trailer skin temperature reaches 96°F.

Table 8. Sample Run 2.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	66.2	69.2	100.0
2	120.0	68.3	69.6	68.3	100.0
3	180.0	70.9	73.0	67.5	100.0
4	240.0	73.4	76.1	66.6	100.0
5	300.0	76.0	79.1	65.8	100.0
6	360.0	78.5	81.7	64.9	100.0
7	420.0	80.0	83.1	64.0	100.0
8	480.0	79.0	81.8	63.0	100.0
9	480.0	78.0	95.2	62.1	100.0
10	480.0	77.0	89.2	61.2	100.0
11	550.0	76.0	77.2	60.1	100.0
12	620.0	75.0	75.8	59.0	100.0
13	690.0	74.1	74.0	57.8	100.0
14	760.0	73.1	72.9	56.6	100.0
15	830.0	72.1	71.9	55.4	100.0
16	900.0	71.1	70.9	54.1	100.0
17	900.0	70.1	68.8	52.7	100.0
18	900.0	69.1	67.8	51.3	100.0
19	900.0	68.1	66.8	49.9	100.0
20	900.0	67.2	65.8	48.5	100.0
21	900.0	66.2	64.8	47.0	100.0
22	900.0	65.2	63.8	45.5	91.6
23	900.0	64.2	62.9	44.3	73.2
24	900.0	63.2	61.9	43.4	59.2

Note: The new number of exposed units is 48. Heat is therefore transferred out of the load more quickly.

Table 9. Sample Run 3.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	66.0	49.6	48.5
2	120.0	68.3	69.5	49.2	48.0
3	180.0	70.9	72.8	48.8	47.6
4	240.0	73.4	76.0	48.4	47.2
5	300.0	76.0	78.9	48.1	46.8
6	360.0	78.5	81.6	47.7	46.4
7	420.0	80.0	82.9	47.4	45.6
8	480.0	79.0	81.7	47.1	43.9
9	480.0	78.0	94.3	46.8	47.2
10	480.0	77.0	88.3	46.6	44.0
11	550.0	76.0	77.1	46.3	39.1
12	620.0	75.0	75.7	46.1	37.7
13	690.0	74.1	73.9	45.9	36.1
14	760.0	73.1	72.8	45.7	34.9
15	830.0	72.1	71.9	45.5	33.8
16	900.0	71.1	70.9	45.3	32.7
17	900.0	70.1	68.4	45.1	31.1
18	900.0	69.1	67.5	45.0	30.1
19	900.0	68.1	66.6	44.8	29.1
20	900.0	67.2	65.6	44.7	28.2
21	900.03	66.2	64.7	44.5	27.3
22	900.0	66.2	63.8	44.4	26.5
23	900.0	65.2	62.9	44.3	25.7
24	900.0	64.2	61.9	44.2	24.9

Note: Product loaded at 50°F instead of 70°F. The chiller therefore does not work as hard.

Table 10. Sample Run 4.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	66.3	71.1	0.0
2	120.0	68.3	69.7	72.4	0.0
3	180.0	70.9	73.1	73.7	0.0
4	240.0	73.4	76.3	75.0	0.0
5	300.0	76.0	79.2	76.5	0.0
6	360.0	78.5	81.8	78.1	0.0
7	420.0	80.0	83.2	79.8	0.0
8	480.0	79.0	82.0	81.6	0.0
9	480.0	78.0	96.7	83.6	0.0
10	480.0	77.0	90.9	85.7	0.0
11	550.0	76.0	77.4	87.8	0.0
12	620.0	75.0	76.0	90.1	0.0
13	690.0	74.1	74.3	92.5	0.0
14	760.0	73.1	73.2	95.2	0.0
15	830.0	72.1	72.2	98.0	0.0
16	900.0	71.1	71.3	101.0	0.0
17	900.0	70.1	71.9	104.3	0.0
18	900.0	69.1	71.1	107.9	0.0
19	900.0	68.1	70.4	111.9	0.0
20	900.0	67.2	69.7	116.3	0.0
21	900.0	66.2	69.0	121.1	0.0
22	900.0	65.2	68.3	126.5	0.0
23	900.0	64.2	67.7	132.6	0.0
24	900.0	63.2	67.1	139.4	0.0

Note: Chiller is shut off (or non-refrigerated trailer). Product temperature reaches 140°F due primarily to respiration.

Table 11. Sample Run 5.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	65.4	69.9	100.0
2	120.0	68.3	68.7	69.8	100.0
3	180.0	70.9	72.0	69.9	100.0
4	240.0	73.4	75.1	70.2	100.0
5	300.0	76.0	77.9	70.5	100.0
6	360.0	78.5	80.5	70.9	100.0
7	420.0	80.0	81.8	71.5	100.0
8	480.0	79.0	80.7	72.0	100.0
9	480.0	78.0	85.7	72.8	100.0
10	550.0	77.0	81.0	73.4	100.0
11	620.0	76.0	76.3	73.8	100.0
12	690.0	75.0	75.0	74.1	100.0
13	760.0	74.1	73.3	74.5	100.0
14	830.0	73.1	72.3	74.7	100.0
15	900.0	72.1	71.3	75.0	100.0
16	900.0	71.1	70.4	75.2	100.0
17	900.0	70.1	65.5	75.2	100.0
18	900.0	69.1	64.7	75.2	100.0
19	900.0	68.1	64.0	75.1	100.0
20	900.0	67.2	63.2	75.1	100.0
21	900.0	66.2	62.4	75.0	100.0
22	900.0	65.2	61.7	74.9	100.0
23	900.0	64.2	60.9	74.7	100.0
24	900.0	63.2	60.1	74.5	100.0

Note: The 3" foam is replaced with 3/4" plywood. Heat from outside therefore gets in more easily (solar and high outside temperature). Chiller works constantly and manages only to maintain loading temperature.

Table 12. Sample Run 6.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	60.0	65.8	66.0	50.5	52.4
2	120.0	68.3	69.5	50.1	51.7
3	180.0	70.9	72.8	49.6	51.1
4	240.0	73.4	76.0	49.2	50.5
5	300.0	76.0	78.9	48.8	49.9
6	360.0	78.5	81.6	48.4	49.3
7	420.0	80.0	82.9	48.1	48.4
8	480.0	79.0	81.7	47.8	46.6
9	480.0	78.0	94.3	47.5	49.7
10	550.0	77.0	88.3	47.2	46.4
11	620.0	76.0	77.1	46.9	41.3
12	690.0	75.0	75.7	46.6	39.8
13	760.0	74.1	73.9	46.4	38.1
14	830.0	73.1	72.8	46.1	36.8
15	900.0	72.1	71.9	45.9	35.5
16	900.0	71.8	70.9	45.7	34.3
17	900.0	70.1	68.4	45.5	32.7
18	900.0	69.1	67.5	45.3	31.6
19	900.0	68.1	66.6	45.2	30.5
20	900.0	67.2	65.6	45.0	29.5
21	900.0	66.2	64.7	44.8	28.6
22	900.0	65.2	63.8	44.7	27.6
23	900.0	64.2	62.9	44.6	26.8
24	900.0	63.2	61.9	44.4	25.9

Note: No changes to the settings from Sample Run 1(Table 7). This is a continuation of the last trip(nest day results). The same speed schedule was used for convenience. The final load temperature on the previous day is used for the new load temperature for the following day.

Table 13. Sample Run 7.

Hours Past Dawn	Distance Travelled (miles)	Air Temperature (°F)	Skin Temperature (°F)	Load Temperature (°F)	% Duty
1	0.0	66.5	72.0	69.4	0.0
2	0.0	69.1	83.9	73.0	0.0
3	0.0	71.6	96.0	80.0	0.0
4	0.0	74.2	106.7	88.7	0.0
5	0.0	76.7	114.9	97.7	0.0
6	0.0	79.3	120.1	105.7	0.0
7	0.0	80.7	122.4	111.9	0.0
8	0.0	79.7	120.4	115.7	0.0
9	0.0	78.7	115.0	116.4	0.0
10	0.0	77.7	106.7	114.1	0.0
11	0.0	76.8	100.8	109.7	0.0
12	0.0	75.8	94.3	104.9	0.0
13	0.0	74.8	82.7	98.3	0.0
14	0.0	73.8	78.8	91.2	0.0
15	0.0	72.8	76.6	85.8	0.0
16	0.0	71.9	74.7	81.8	0.0
17	0.0	70.9	73.1	78.7	0.0
18	0.0	69.9	71.7	76.2	0.0
19	0.0	68.9	70.4	74.1	0.0
20	0.0	67.9	69.2	72.4	0.0
21	0.0	66.9	68.1	70.9	0.0
22	0.0	66.0	67.0	69.6	0.0
23	0.0	65.0	65.9	68.3	0.0
24	0.0	64.0	64.9	67.2	0.0

Note: New product (15,000 lbs of steel). No refrigeration, no respiration. Trailer with aluminum walls only. Cloud coverage is zero. The trailer is stationary in Los Angeles but points toward Cincinnati (speeds all zero). Skin temperature reaches 122°F. Load temperature reaches 116°F.

APPENDIX B

Source Code of PACKTEMP in 'C'

```
#include <windows.h>

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include "r2.h"

extern void uslines(HDC);

void calc(void);

long FAR PASCAL WndProc (HWND, WORD, WORD, LONG);

HANDLE hInst;

int flag=0;

int a0,a1,a2,a3,a4=0;

int OutOfReach;

int r40=550;

int r60=2405;

int r80=6005;

char *cr80="6005";

double fr40,fr60,fr80;

double fhc=0.72; /*float hc */

int hc=72; /*product specific heat*/

char outarr[28][100];

int lenstr[24];

int speed[24]={60, 60, 60, 60, 60, 60, 60, 60, 0, 0,

                    70, 70, 70, 70, 70, 70, 0, 0, 0, 0,

                    0, 0, 0, 0};
```

```
struct userdat  {       int startcity;
                        int destcity;
                        int dayofyear;
                        int rh;
                        int cloud;
                        int prodwt;
                        int expunits;
                        int prodtype;
                        int loadtemp;
                        int thermoset;
                };
struct trailer  {  int length;
                        int width;
                        int height;
                        int btu;
                        int hw; /*wall heat transfer*/
                };
struct citytemp {       char *abv;
                        char *cityname;
                        double hia;
                        double hib;
                        double lowa;
                        double lowb;
                        double lati;
                        double longi;
                        int scrlati;
```

```
                int scrlongl;
        };
struct prodinfo { char *prodname;
                double rateca;
                double ratecb;
                double heatc;};
struct citytemp  ctemparr[]={
    "AMA","AMARILLO",       49.9, 45.0, 23.6, 43.3, 35.2, 101.8,-223,267,
    "AVL","ASHVILLE",       50.2, 35.8, 28.4, 35.1, 35.4, 82.5, -226,473,
    "AST","ASTORIA",        48.1, 21.0, 34.7, 17.4, 46.2, 123.8,-393,33,
    "ATL","ATLANTA",        54.1, 35.4, 34.0, 34.6, 33.7, 84.4, -199,453,
    "BIS","BISMARK",        22.1, 64.5, 3.80, 54.7, 46.8, 100.8,-409,278,
    "BOI","BOISE",          37.4, 52.1, 22.7, 33.1, 43.6, 116.2,-358,114,
    "BOS","BOSTON",         37.1, 44.6, 23.6, 40.6, 42.4, 71.0, -338,596,
    "BRV","BROWNSVILLE",        71.6, 22.2, 53.2, 24.1, 25.9, 97.5, -74,313,
    "BUF","BUFFALO",        30.9, 50.1, 17.5, 41.3, 42.9, 78.9, -346,511,
    "BTV","BURLINGTON",     27.7, 55.1, 11.1, 48.2, 44.5, 73.2, -372,572,
    "CBU","CARIBOU",        20.9, 54.8, 4.30, 50.5, 46.9, 68.0, -410,618,/*M 628 */
    "CHS","CHARLESTON",     61.0, 30.2, 37.2, 35.1, 32.9, 80.0, -186,500,
    "ORD","CHICAGO",        33.4, 51.6, 19.7, 47.4, 41.8, 87.8, -329,416,
    "CVG","CINCINNATI",     43.0, 47.0, 27.2, 41.8, 39.1, 84.5, -286,452,
    "CMH","COLUMBUS",       38.0, 50.1, 20.9, 41.6, 40.0, 83.0, -300,468,
    "CRL","CRATER LAKE",    32.8, 34.4, 16.9, 22.9, 42.9, 122.1,-346,51,
    "DEN","DENVER",         41.6, 45.7, 14.7, 41.1, 39.7, 104.9,-295,234,
    "DSM","DES MOINES",     31.3, 58.2, 14.3, 53.1, 41.5, 93.7, -324,354,
    "DTW","DETROIT",        33.5, 53.2, 19.5, 42.8, 42.4, 83.0, -338,468,
```

"EPS","EL PASO", 57.9, 39.0, 30.1, 39.4, 31.8, 106.5,-169,217,

"EUR","EUREKA", 54.4, 6.40, 41.6, 10.8, 40.8, 124.2,-313,29,

"FAR","FARGO", 19.0, 66.6, 2.40, 57.4, 46.9, 96.8, -410,321,

"FSF","FLAGSTAFF", 41.0, 40.1, 14.0, 33.6, 35.2, 111.7,-223,162,

"FTW","FORT WORTH", 57.0, 39.6, 35.2, 40.3, 32.8, 97.3, -185,315,

"FAT","FRESNO", 56.9, 40.6, 36.4, 24.3, 36.7, 119.8,-247,75,

"GLA","GLACIER", 28.7, 50.4, 15.3, 31.4, 48.6, 113.7,-438,141,

"GRJ","GRAND JUNCTION", 37.4, 55.1, 13.6, 44.7, 39.0, 108.6,-284,195,

"GRB","GREEN BAY", 25.1, 56.8, 10.4, 49.9, 44.5, 88.0, -372,415,

"HAT","HATTERAS", 55.2, 29.2, 39.4, 33.1, 35.3, 75.6, -225,547,

"HAV","HAVRE", 27.0, 58.2, 4.70, 48.7, 48.5, 109.7,-436,183,

"HOU","HOUSTON", 64.5, 29.6, 44.1, 20.8, 30.0, 95.4, -140,336,

"INL","INTERNATIONAL FALLS",16.3, 63.1, -4.0, 58.8, 48.6, 93.4, -438,357,

"JAX","JACKSONVILLE", 66.7, 25.4, 44.8, 29.4, 30.5, 81.7, -148,482,

"LND","LANDER", 32.1, 52.1, 9.10, 45.4, 42.8, 108.7,-345,194,

"LAS","LAS VEGAS", 54.8, 48.4, 31.8, 42.4, 36.1, 115.2,-238,125,

"LNK","LINCOLN", 36.3, 55.5, 17.1, 50.1, 40.8, 96.7, -313,322,

"LIT","LITTLE ROCK", 51.4, 43.1, 30.5, 41.0, 34.7, 92.2, -215,370,

"LAX","LOS ANGELES", 65.1, 9.50, 46.2, 15.3, 33.9, 118.4,-202,90,

"MEM","MEMPHIS", 52.9, 40.5, 32.5, 39.4, 35.0, 90.0, -220,393,

"MIA","MIAMI", 76.6, 13.2, 58.7, 17.5, 25.8, 80.3, -73,497,

"MID","MIDLAND", 57.1, 39.8, 31.2, 41.0, 32.0, 102.0,-172,265,

"MCT","MILES CITY", 29.5, 59.3, 8.00, 50.4, 46.4, 105.8,-402,225,

"MSP","MINNEAPOLIS", 23.9, 62.4, 4.90, 57.2, 44.9, 93.2, -378,359,

"MSO","MISSOULA", 29.4, 54.7, 12.9, 35.5, 46.9, 114.0,-410,137,

"MGM","MONTGOMERY", 58.6, 35.2, 36.5, 35.6, 32.4, 86.3, -178,433,

```
"BNA","NASHVILLE",      49.2, 43.0, 29.9, 39.4, 36.2, 86.8, -239,427,
"MSY","NEW ORLEANS",        64.4, 27.9, 45.0, 28.7, 30.0, 90.3, -140,390,
"JFK","NEW YORK",       39.6, 45.4, 27.1, 41.0, 40.8, 74.0, -313,564,
"ORF","NORFOLK",        49.9, 38.6, 31.1, 39.3, 36.9, 76.3, -250,539,
"NPL","NORTH PLATTE",       37.6, 51.5, 10.7, 50.2, 41.1, 100.8,-318,278,
"OKC","OKLAHOMA CITY",      48.0, 44.8, 27.3, 43.8, 35.4, 97.6, -226,312,
"PHL","PHILADELPHIA",       40.6, 46.2, 23.5, 41.2, 39.9, 75.3, -298,550,
"PIT","PITTSBURGH",     36.7, 48.2, 20.1, 41.0, 40.5, 80.2, -308,498,
"PDX","PORTLAND",       45.6, 32.9, 33.4, 21.5, 45.6, 122.6,-380,46,/*M-390*/
"RDU","RALEIGH",        52.3, 37.3, 29.4, 38.3, 35.8, 78.6, -233,515,
"RAP","RAPID CITY",     34.0, 51.8, 10.6, 48.3, 44.1, 103.0,-366,255,
"RNO","RENO",           46.2, 41.4, 16.6, 27.3, 39.5, 119.8,-292,75,
"SLC","SALT LAKE CITY",     37.7, 52.5, 19.6, 38.9, 40.8, 112.0,-313,159,
"SAT","SAN ANTONIO", 63.6, 31.6, 42.0, 32.9, 29.4, 98.5, -130,303,
"SFO","SAN FRANCISCO",      57.8, 15.6, 42.6, 11.7, 37.6, 122.4,-262,48,
"SSM","SAULT STE MARIE",    23.4, 52.6, 8.70, 44.3, 46.5, 84.3, -404,454,
"SEA","SEATTLE",        44.0, 30.2, 33.2, 19.9, 47.5, 122.3,-420,49,
"SHV","SHREVEPORT",  57.8, 37.0, 38.0, 35.9, 32.5, 93.8, -180,353,
"GEG","SPOKANE",        32.9, 49.9, 20.8, 33.1, 47.7, 117.4,-423,101,
"STL","ST LOUIS",       41.7, 48.5, 23.4, 43.8, 38.8, 90.4, -281,389,
"TLH","TALLAHASSEE", 65.7, 27.0, 41.8, 31.2, 30.5, 84.3, -148,454,
"TPA","TAMPA",          72.0, 19.4, 51.4, 23.2, 28.0, 82.5, -108,473,
"DCA","WASHINGTON", 44.6, 43.2, 28.3, 40.7, 38.9, 77.0, -282,532,
"ICT","WICHITA",        45.1, 48.2, 22.9, 46.7, 37.7, 97.3, -263,315,
"WIL","WILLISTON",      21.9, 12.3, 3.70, 55.4, 48.1, 103.6,-430,248,
"YST","YELLOW STONE",       27.7, 49.3, 10.3, 34.8, 45.0, 110.7,-380,173,
```

```
        "YUM","YUMA",          70.7, 37.8, 41.3, 36.3, 32.7, 114.6,-183,131
};
struct userdat userinfo={38,14,227,90,50,2500,1,34,70,40};
struct trailer trailerdef={48,8,8,435,8};
struct prodinfo prodarr[]=
{       "Aluminum",     0,      0,      0.21,
        "Apples",       0.253,  0.057,  0.87,
        "Artichokes",   1.162,  0.041,  0.87,
        "Asparagus",    1.820,  0.043,  0.94,
        "Bacon",        0,      0,      0.38,
        "Beef",         0,      0,      0.77,
        "Blueberries",  0.555,  0.060,  0.86,
        "Brick", 0,     0,      0.20,
        "Brocolli",     3.433,  0.065,  0.92,
        "Cantiloup",    0.386,  0.055,  0.94,
        "Cauliflower",  0.512,  0.040,  0.93,
        "Cheese",       0,      0,      0.52,
        "Copper",       0,      0,      0.09,
        "Corn(Sweet)",  1.858,  0.047,  0.79,
        "Eggs",         0,      0,      0.73,
        "Glass",        0,      0,      0.20,
        "Grapes",       0.216,  0.056,  0.86,
        "Lettuce",      0.385,  0.041,  0.96,
        "Liquid Chemicals",0,   0,      0.5,
        "Milk",         0,      0,      0.93,
        "Green Onions",  0.803,  0.043,  0.90,
```

```
"Oranges",      0.185,  0.05,   0.9,
"Peaches",      0.58,   0.067,  0.91,
"Pears",        0.504,  0.65,   0.86,
"Peas",         1.985,  0.48,   0.79,
"Plastic",      0,      0,      0.5,
"Plums",        0.197,  0.055,  0.88,
"Pork",         0,      0,      0.53,
"Potatoes",     0.25,   0.046,  0.85,
"Poultry",      0,      0,      0.8,
"Radishes",     0.378,  0.053,  0.95,
"Spinach",      2.392,  0.064,  0.94,
"Steel",        0,      0,      0.11,
"Strawberries", 1.027,  0.055,  0.92,
"Tin",          0,      0,      0.05,
"Watermelon",   0.385,  0.06,   0.97,
"Wood",         0,      0,      0.67 };

int PASCAL WinMain  (HANDLE hInstance,
                     HANDLE hPrevInstance,
                     LPSTR lpszCmdParam,
                     int nCmdShow)
  {
  static      char szAppName[] = "R2";
  HWND        hwnd;
  MSG         msg;
  WNDCLASS    wndclass;
```

```
if(!hPrevInstance)
   {
   wndclass.style            = CS_HREDRAW | CS_VREDRAW;
   wndclass.lpfnWndProc      = WndProc;
   wndclass.cbClsExtra       = 0;
   wndclass.cbWndExtra       = 0;
   wndclass.hInstance        = hInstance;
   wndclass.hIcon            = LoadIcon(hInstance, szAppName);
   wndclass.hCursor          = LoadCursor(hInstance, szAppName);
   wndclass.hbrBackground    = GetStockObject(WHITE_BRUSH);
   wndclass.lpszMenuName     = szAppName;
   wndclass.lpszClassName    = szAppName;
   RegisterClass(&wndclass);
   }
hwnd = CreateWindow (szAppName,
                    "PACKTEMP",
                    WS_OVERLAPPEDWINDOW,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    CW_USEDEFAULT,
                    NULL,
                    NULL,
                    hInstance,
                    NULL);
```

```
ShowWindow(hwnd,nCmdShow);
UpdateWindow(hwnd);
while(GetMessage (&msg, NULL, 0, 0))
  {
  TranslateMessage(&msg);
  DispatchMessage(&msg);
  }
  return msg.wParam;
}
BOOL FAR PASCAL DeftDlgProc(HWND hDlg, WORD message, WORD wParam, LONG lParam)
{
  switch (message)
  { case WM_INITDIALOG:
        SetDlgItemInt(hDlg,20,trailerdef.length,0);
        SetDlgItemInt(hDlg,21,trailerdef.width, 0);
        SetDlgItemInt(hDlg,22,trailerdef.height,0);
        SetDlgItemInt(hDlg,23,trailerdef.btu,   0);
        SetDlgItemInt(hDlg,99,trailerdef.hw,0);
        SetDlgItemInt(hDlg,31,speed[0],0);
        SetDlgItemInt(hDlg,32,speed[1],0);
        SetDlgItemInt(hDlg,33,speed[2],0);
        SetDlgItemInt(hDlg,34,speed[3],0);
        SetDlgItemInt(hDlg,35,speed[4],0);
        SetDlgItemInt(hDlg,36,speed[5],0);
        SetDlgItemInt(hDlg,37,speed[6],0);
        SetDlgItemInt(hDlg,38,speed[7],0);
```

```
SetDlgItemInt(hDlg,39,speed[8],0);
SetDlgItemInt(hDlg,40,speed[9],0);
SetDlgItemInt(hDlg,41,speed[10],0);
SetDlgItemInt(hDlg,42,speed[11],0);
SetDlgItemInt(hDlg,43,speed[12],0);
SetDlgItemInt(hDlg,44,speed[13],0);
SetDlgItemInt(hDlg,45,speed[14],0);
SetDlgItemInt(hDlg,46,speed[15],0);
SetDlgItemInt(hDlg,47,speed[16],0);
SetDlgItemInt(hDlg,48,speed[17],0);
SetDlgItemInt(hDlg,49,speed[18],0);
SetDlgItemInt(hDlg,50,speed[19],0);
SetDlgItemInt(hDlg,51,speed[20],0);
SetDlgItemInt(hDlg,52,speed[21],0);
SetDlgItemInt(hDlg,53,speed[22],0);
SetDlgItemInt(hDlg,54,speed[23],0);
return TRUE;
case WM_COMMAND:
switch(wParam)
  { case IDOK:
trailerdef.length=GetDlgItemInt(hDlg,20,NULL,1);
trailerdef.width =GetDlgItemInt(hDlg,21,NULL,1);
trailerdef.height=GetDlgItemInt(hDlg,22,NULL,1);
trailerdef.btu   =GetDlgItemInt(hDlg,23,NULL,1);
trailerdef.hw      =GetDlgItemInt(hDlg,99,NULL,1);
speed[0]           =GetDlgItemInt(hDlg,31,NULL,1);
```

```
speed[1]        =GetDlgItemInt(hDlg,32,NULL,1);
speed[2]        =GetDlgItemInt(hDlg,33,NULL,1);
speed[3]        =GetDlgItemInt(hDlg,34,NULL,1);
speed[4]        =GetDlgItemInt(hDlg,35,NULL,1);
speed[5]        =GetDlgItemInt(hDlg,36,NULL,1);
speed[6]        =GetDlgItemInt(hDlg,37,NULL,1);
speed[7]        =GetDlgItemInt(hDlg,38,NULL,1);
speed[8]        =GetDlgItemInt(hDlg,39,NULL,1);
speed[9]        =GetDlgItemInt(hDlg,40,NULL,1);
speed[10]       =GetDlgItemInt(hDlg,41,NULL,1);
speed[11]       =GetDlgItemInt(hDlg,42,NULL,1);
speed[12]       =GetDlgItemInt(hDlg,43,NULL,1);
speed[13]       =GetDlgItemInt(hDlg,44,NULL,1);
speed[14]       =GetDlgItemInt(hDlg,45,NULL,1);
speed[15]       =GetDlgItemInt(hDlg,46,NULL,1);
speed[16]       =GetDlgItemInt(hDlg,47,NULL,1);
speed[17]       =GetDlgItemInt(hDlg,48,NULL,1);
speed[18]       =GetDlgItemInt(hDlg,49,NULL,1);
speed[19]       =GetDlgItemInt(hDlg,50,NULL,1);
speed[20]       =GetDlgItemInt(hDlg,51,NULL,1);
speed[21]       =GetDlgItemInt(hDlg,52,NULL,1);
speed[22]       =GetDlgItemInt(hDlg,53,NULL,1);
speed[23]       =GetDlgItemInt(hDlg,54,NULL,1);
        EndDialog(hDlg,FALSE);/*TRUE*/
        return TRUE;
case 26:
```

```
trailerdef.length=GetDlgItemInt(hDlg,20,NULL,1);
trailerdef.width =GetDlgItemInt(hDlg,21,NULL,1);
trailerdef.height=GetDlgItemInt(hDlg,22,NULL,1);
trailerdef.btu   =GetDlgItemInt(hDlg,23,NULL,1);
trailerdef.hw    =GetDlgItemInt(hDlg,99,NULL,1);
speed[0]         =GetDlgItemInt(hDlg,31,NULL,1);
speed[1]         =GetDlgItemInt(hDlg,32,NULL,1);
speed[2]         =GetDlgItemInt(hDlg,33,NULL,1);
speed[3]         =GetDlgItemInt(hDlg,34,NULL,1);
speed[4]         =GetDlgItemInt(hDlg,35,NULL,1);
speed[5]         =GetDlgItemInt(hDlg,36,NULL,1);
speed[6]         =GetDlgItemInt(hDlg,37,NULL,1);
speed[7]         =GetDlgItemInt(hDlg,38,NULL,1);
speed[8]         =GetDlgItemInt(hDlg,39,NULL,1);
speed[9]         =GetDlgItemInt(hDlg,40,NULL,1);
speed[10]        =GetDlgItemInt(hDlg,41,NULL,1);
speed[11]        =GetDlgItemInt(hDlg,42,NULL,1);
speed[12]        =GetDlgItemInt(hDlg,43,NULL,1);
speed[13]        =GetDlgItemInt(hDlg,44,NULL,1);
speed[14]        =GetDlgItemInt(hDlg,45,NULL,1);
speed[15]        =GetDlgItemInt(hDlg,46,NULL,1);
speed[16]        =GetDlgItemInt(hDlg,47,NULL,1);
speed[17]        =GetDlgItemInt(hDlg,48,NULL,1);
speed[18]        =GetDlgItemInt(hDlg,49,NULL,1);
speed[19]        =GetDlgItemInt(hDlg,50,NULL,1);
speed[20]        =GetDlgItemInt(hDlg,51,NULL,1);
```

```
        speed[21]         =GetDlgItemInt(hDlg,52,NULL,1);
        speed[22]         =GetDlgItemInt(hDlg,53,NULL,1);
        speed[23]         =GetDlgItemInt(hDlg,54,NULL,1);
calc();
flag=1;
        EndDialog(hDlg,TRUE);
        return TRUE;
          case IDCANCEL:
                EndDialog(hDlg,FALSE);
                return TRUE;
        break;
        }
   break;
}
  return FALSE;
}
BOOL FAR PASCAL AboutDlgProc(HWND hDlg, WORD message, WORD wParam, LONG lParam)
{
  switch (message)
  { case WM_INITDIALOG:
        SetDlgItemInt(hDlg,10,userinfo.startcity, 1);
        SetDlgItemInt(hDlg,11,userinfo.destcity,  1);
        SetDlgItemInt(hDlg,12,userinfo.dayofyear, 1);
        SetDlgItemInt(hDlg,13,userinfo.rh,        1);
        SetDlgItemInt(hDlg,14,userinfo.cloud,     1);
        SetDlgItemInt(hDlg,15,userinfo.prodwt,    1);
```

```
        SetDlgItemInt(hDlg,16,userinfo.expunits,  1);
/*      SetDlgItemInt(hDlg,17,userinfo.prodtype,  1);
*/      SetDlgItemInt(hDlg,18,userinfo.loadtemp,  1);
        SetDlgItemInt(hDlg,19,userinfo.thermoset, 1);
        SetDlgItemInt(hDlg,71,r40,1);
        SetDlgItemInt(hDlg,72,r60,1);
        SetDlgItemInt(hDlg,73,r80,1);
        SetDlgItemInt(hDlg,74,hc,1);
        return TRUE;


        case WM_COMMAND:
        switch(wParam)
         {
        case IDOK:
          userinfo.startcity       =GetDlgItemInt(hDlg,10,NULL,1);
          userinfo.destcity        =GetDlgItemInt(hDlg,11,NULL,1);
          userinfo.dayofyear       =GetDlgItemInt(hDlg,12,NULL,1);
          userinfo.rh              =GetDlgItemInt(hDlg,13,NULL,1);
          userinfo.cloud           =GetDlgItemInt(hDlg,14,NULL,1);
          userinfo.prodwt          =GetDlgItemInt(hDlg,15,NULL,1);
          userinfo.expunits        =GetDlgItemInt(hDlg,16,NULL,1);
          userinfo.prodtype        =GetDlgItemInt(hDlg,17,NULL,1);
          userinfo.loadtemp        =GetDlgItemInt(hDlg,18,NULL,1);
          userinfo.thermoset       =GetDlgItemInt(hDlg,19,NULL,1);
          hc                       =GetDlgItemInt(hDlg,74,NULL,1);
          fhc                      =hc/100.0;
```

```
  r40                   =GetDlgItemInt(hDlg,71,NULL,1);
  r60                   =GetDlgItemInt(hDlg,72,NULL,1);
  r80                   =GetDlgItemInt(hDlg,73,NULL,1);
  EndDialog(hDlg,FALSE);
return TRUE;

case 25:
  userinfo.startcity    =GetDlgItemInt(hDlg,10,NULL,1);
  userinfo.destcity     =GetDlgItemInt(hDlg,11,NULL,1);
  userinfo.dayofyear    =GetDlgItemInt(hDlg,12,NULL,1);
  userinfo.rh           =GetDlgItemInt(hDlg,13,NULL,1);
  userinfo.cloud        =GetDlgItemInt(hDlg,14,NULL,1);
  userinfo.prodwt       =GetDlgItemInt(hDlg,15,NULL,1);


  userinfo.expunits     =GetDlgItemInt(hDlg,16,NULL,1);
  userinfo.prodtype     =GetDlgItemInt(hDlg,17,NULL,1);
  userinfo.loadtemp     =GetDlgItemInt(hDlg,18,NULL,1);
  userinfo.thermoset    =GetDlgItemInt(hDlg,19,NULL,1);
  hc                    =GetDlgItemInt(hDlg,74,NULL,1);
  fhc                   =hc/100.0;
  r40                   =GetDlgItemInt(hDlg,71,NULL,1);
  r60                   =GetDlgItemInt(hDlg,72,NULL,1);
  r80                   =GetDlgItemInt(hDlg,73,NULL,1);
  calc();
  flag=1;
```

```
            EndDialog(hDlg,TRUE);
           return TRUE;

           case IDCANCEL:
            EndDialog(hDlg,FALSE);
           return TRUE;
           }
    break;
}
  return FALSE;
}
long FAR PASCAL WndProc(HWND hwnd, WORD message, WORD wParam, LONG lParam)
{
static FARPROC      lpfnAboutDlgProc;
static FARPROC      lpfnDeftDlgProc;
static HANDLE       hInstance;
static FARPROC      lpfnLineProc;

HDC                 hdc;
PAINTSTRUCT         ps;
RECT                rect;
HRGN                hRgn;
POINT               point;
HMENU               hMenu;
HBRUSH              hBrush;
HFONT               hFont;
```

```
char                    szBuffer[40];
TEXTMETRIC              tm;
int                     citycnt,strcount,xpos,ypos;
static short            cxClient, cyClient, cxChar,cyChar,cxCaps;
citycnt=0;
switch(message)
{
 case WM_CREATE:
        hInstance=((LPCREATESTRUCT)lParam)->hInstance;
        lpfnAboutDlgProc=MakeProcInstance(AboutDlgProc,hInstance);
        lpfnDeftDlgProc =MakeProcInstance(DeftDlgProc, hInstance);
        hdc=GetDC(hwnd);
        GetTextMetrics(hdc,&tm);
        cxChar=tm.tmAveCharWidth;
        cxCaps=(tm.tmPitchAndFamily & 1 ? 3: 2) * cxChar /2 ;
        cyChar=tm.tmHeight + tm.tmExternalLeading;
        ReleaseDC(hwnd,hdc);
        return 0;
case WM_COMMAND:
        hMenu = GetMenu(hwnd);
        switch(wParam)
        {
        case IDM_ABOUT:
                DialogBox(hInstance,"AboutBox",hwnd,lpfnAboutDlgProc);
                InvalidateRect(hwnd,NULL,TRUE);
                return 0;
```

```
        case IDM_DEF:
                DialogBox(hInstance,"Deft",hwnd,lpfnDeftDlgProc);
                InvalidateRect(hwnd,NULL,TRUE);
                return 0;
        }
        break;
      return 0;
case WM_LBUTTONDOWN:
        return 0;
case WM_RBUTTONDOWN:
        flag=2;
        InvalidateRect(hwnd,NULL,TRUE);
        return 0;
case WM_MOUSEMOVE:
        return 0;
case WM_PAINT:
  hdc = BeginPaint(hwnd, &ps);
  SetMapMode(hdc,MM_TEXT);
  SetViewportOrg(hdc,0,480);
  SelectObject(hdc,GetStockObject(ANSI_VAR_FONT));
  uslines(hdc);
  SetTextColor(hdc,RGB(100,100,250));

for (citycnt=0;citycnt<73;citycnt++)
        TextOut(hdc,ctemparr[citycnt].scrlongi,ctemparr[citycnt].scrlati,
        ctemparr[citycnt].abv,3);
```

```
SetRect(&rect,100,-450,500,-20);
if (flag==1){
        FillRect(hdc,&rect,GetStockObject(LTGRAY_BRUSH));
        TextOut(hdc,120,-430,outarr[0],a0);
if(OutOfReach==0)
{       TextOut(hdc,120,-417,outarr[1],a1);
        TextOut(hdc,120,-404,outarr[2],a2);
        TextOut(hdc,120,-391,outarr[3],a3);
        for (strcount=0;strcount<24;strcount++)
        {TextOut(hdc,120,-1*(365-13*strcount),outarr[strcount+4],lenstr[strcount]);
        }
}
        SelectObject(hdc,GetStockObject(WHITE_BRUSH));
        flag=0;
        OutOfReach=0;
        }
if (flag==2) flag=0;


EndPaint(hwnd,&ps);
return 0;
case WM_DESTROY:
          PostQuitMessage(0);
          return 0;
}
return DefWindowProc(hwnd, message, wParam, lParam);
}
```

```
void calc(void)
{
int i,hFile;
double nu, day, lat1, long1, lat2, long2, ah1, bh1, al1, bl1, ah2, bh2, al2,
bl2, rh, cc, ln, wd, ht, btu, wt, a, b, tl, tset, vel;
int hour;
double lat, tmin, tmax, dist, tim, sv, cv, st, ct, sd, cd, sl, cl,
zen, r, dawn, dusk, lday, ao, al, sh, ch, sa,
insol, top, ff, fr, front, right, fb, fl, back, left, solar, te,
heq,  duty, ta, ts, p, q, k1, k2, hi, ho, f;
double tes,tes2;
double    h;
double  h1,h2;
double rsp;
/*float r40,r60,r80;Product heat coef at various temp*/
double hw;  /*Wall Heat Transfer Coef*/
double tl1,tl2,th1,th2; /*NEW VARS FOR TEMP RECALC*/
double dtrv;
double c1,c2,c3; /*Line 500 of BASIC*/
static char *szFileName="test.txt";
static POFSTRUCT pof;
lat1 =    ctemparr[userinfo.startcity-1].lati;
long1 = ctemparr[userinfo.startcity-1].longi;
lat2 =    ctemparr[userinfo.destcity-1].lati;
long2 = ctemparr[userinfo.destcity-1].longi;
day =    userinfo.dayofyear;
```

```
ah1 =   ctemparr[userinfo.startcity-1].hia;
bh1 =   ctemparr[userinfo.startcity-1].hib;
al1 =   ctemparr[userinfo.startcity-1].lowa;
bl1 =   ctemparr[userinfo.startcity-1].lowb;
ah2 =   ctemparr[userinfo.destcity-1].hia;
bh2 =   ctemparr[userinfo.destcity-1].hib;
al2 =   ctemparr[userinfo.destcity-1].lowa;
bl2 =   ctemparr[userinfo.destcity-1].lowb;
rh  =   userinfo.rh;
cc  =   userinfo.cloud;
ln  =   trailerdef.length;
wd  =   trailerdef.width;
ht  =   trailerdef.height;
btu =   trailerdof.btu*100.0;
hw  =   trailerdef.hw/100.0;
wt  =   userinfo.prodwt*10.0;
nu  =   userinfo.expunits;
a   =   prodarr[userinfo.prodtype-1].rateca;
b   =   prodarr[userinfo.prodtype-1].ratecb;
tl  =   userinfo.loadtemp;
tset =  userinfo.thermoset;
tl1=al1+bl1* pow( sin(3.14159*(day-16)/360),2);
tl2=al2+bl2* pow( sin(3.14159*(day-16)/360),2);
th1=ah1+bh1* pow( sin(3.14159*(day-16)/360),2);
th2=ah2+bh2* pow( sin(3.14159*(day-16)/360),2);
for (i=0;i<28;i++)  *outarr[i]='\0';
```

```
OutOfReach=0;

lat=(lat1+lat2)/2;

if ( fabs(lat) <= 66.5 )

    {

    day=(day+10.5)*360.0/365.0;

p        = ( cos(lat1*3.14159/180) + cos(lat2*3.14159/180) )/2;

q        = pow( pow((long2-long1)*p,2) + pow((lat2-lat1),2) , 0.5);

tes2     = q;

dist     = 1.15 *3950 * q * 3.14159 / 180;

if (q<=0)

 { sv=1; cv=0;}

else

 { sv=(long1-long2)*p/q;

   cv=(lat2-lat1)/q;

 }

        st=sin(23.45*3.14159/180);

        ct=cos(23.45*3.14159/180);

        cd=cos(day * 3.14159/180);

        sd=sin(day * 3.14159/180);

        sl=sin(lat*3.14159/180);

        cl=cos(lat*3.14159/180);

        if (sd==0) sd=0.001;

        zen=atan(-ct*cd/sd)*180/3.14159;

        if (sd>0) zen=zen+180;

        p=sqrt( pow(cl,2)- pow(st*cd,2) );

        q=st*sl*cd;
```

```
        if (q==0) q=0.001;
        r=atan(p/q) * 180 / 3.14159;
        if (r<0) r=r+180;
        dawn=zen-r;
        dusk=zen+r;
        lday=(dusk-dawn)/15;
        tmin=(tl1+tl2)/2;
        tmax=(th1+th2)/2;
        if (btu>0)
          btu=btu/(1+0.25*rh/100.0);
          ao=2*(ln*wd+ln*ht+wd*ht);
          al=6* pow((ln*wd*ht),0.6666666) * pow(nu,0.3333333);
fr40=r40*10.0;
fr60=r60*10.0;
fr80=r80*10.0;
          c1= 6*fr40 - 8*fr60 + 3*fr80;
          c2= 7*fr40 - 12*fr60+ 5*fr80;
          c3= 2*fr40 - 4*fr60 + 2*fr80;
          dtrv=0.0;
a0=sprintf(outarr[0],"Available daylight = %3.2f hrs, Dist= %3.2f",lday,dist);
a1=sprintf
  (outarr[1],"Expected min and max temp are %3.2fF and %3.2fF", tmin,tmax);
a2=sprintf(outarr[2],"Travelling from %s to %s",
                ctemparr[userinfo.startcity-1].cityname,
                ctemparr[userinfo.destcity-1].cityname);
```

```
a3=
sprintf(outarr[3],"Hrs Past Dawn     DIST TRAV    Air Temp     Skin Temp        Load Temp
%Duty");
        for (hour=1;hour<25;hour++)
         {
         dtrv=dtrv+speed[hour-1];  /*check subscrpts for speed[hour-1]*/
         h1= (dawn+(hour-1)*15+1);
         h2= h1+14;

           for (h=h1;h<h2+1;h++)
              { sh=sin(h*3.14159/180);
               ch=cos(h*3.14159/180);
               sa=(ct*cl*sh-st*sl)*cd-cl*ch*sd;
               insol=0;
               if (sa>0) insol=442* pow(0.81,(1/sa)) *(1.1-cc/100)* 0.5;
               top=insol*sa*ln*wd;
               ff=(ct*(ch*sv-sl*sh*cv)-st*cl*cv)*cd+
                  (sh*sv+sl*ch*cv)*sd;
               front=0;
               if (ff>0) front=insol*ff*wd*ht;
               fr=(ct*(ch*cv+sl*sh*sv)+st*cl*sv)*cd
                  +(sh*cv-sl*ch*sv)*sd;
               right=0;
               if (fr>0) right=insol*fr*ln*ht;
               fb=-1 * ff;
               back=0;
```

```
            if (fb>0) back=insol*fb*wd*ht;
            fl=-1 * fr;
            left=0;
            if (fl>0) left=insol*fl*ln*ht;
            solar=top+front+right+back+left;
/*outside airtemp, heat trans coeff, resp */
            te=tmin+(tmax-tmin)*(h-dawn)/(zen-dawn);
            if (n>zen)
                   te=tmax-(tmax-tmin)*(h-zen)/(dawn+360-zen);
            hi=1.15;
            ho=1.27+9* pow((speed[hour-1]*1.0/65.0),0.8);
            heq= 1/ ( 1/ho + 1/hw + 1/hi);
            f=heq/ho;
            rsp=(c1 - c2*(tl/40.0) + c3*pow((tl/40.0),2)) /2000.0;
            rsp=rsp/24.0;
/*assuming heater/chiller not on*/
            ta=(f*solar+heq*ao*te+hi*al*tl)/(heq*ao+hi*al);
            ts=f*ta+(1-f)*(te+solar/(ho*ao));
            duty=0;
      if ( (btu==0) || ((btu>0) && (ta<tset)) || ((btu<0)&&(ta>tset)))
              { k1=(rsp*wt)/(hi*al);
                k2=(hi*al)/(wt*fhc); /*divide by 0*/
                tl=ta+k1+(tl-ta-k1)*exp(-k2/15);
              }
            else
              {ta=tset;
```

```
                ts=f*ta+(1-f)*(te+(solar/(ho*ao)));

                q=hi*al*(tl-ta)+f*solar+heq*ao*(te-ta);

                duty=100*q/btu;

                if (duty<100)

                  {

                    k1=(rsp*wt)/(hi*al);

                    k2=(hi*al)/(wt*fhc);

                    tl=ta+k1+(tl-ta-k1)*exp(-k2/15);

                  }

                else{  ta=(f*solar+heq*ao*te+hi*al*tl-btu) /(heq*ao+hi*al);

                        ts=f*ta+(1-f)*(te+solar/(ho*ao));

                        duty=100;

                        k1=(rsp*wt)/(hi*al);

                        k2=(hi*al)/(wt*fhc);

                        tl=ta+k1+(tl-ta-k1)*exp(-k2/15);

                     } /*end else*/

                }/*end else*/

        }/*new end for */

lenstr[hour-1]=

sprintf(outarr[hour+3],"          %2.0d                 %4.1f          %4.1f          %4.1f

 %4.1f     %4.1f\n",hour,dtrv,te,ts,tl,duty);

        /*end for*/

}/*end for*/

}/*end if lat<65 */

}/*end calc*/
```

APPENDIX C

CDP Program in BASIC

```
10 DIM VEL(24)
20 PRINT "enter latitude & longitude of start point in degrees":INPUT LAT1,LONG1
30 PRINT "enter latitude & longitude of destination in degrees":INPUT LAT2,LONG2
40 PRINT"enter departure day of year from Jan 1 (Jan1=1)":INPUT DAY
50 PRINT"enter low & high temperatures at start point in deg F":INPUT TL1,TH1
60 PRINT"enter low & high temperatures at destination in deg F":INPUT TL2,TH2
70 PRINT"enter the expected relative humidity in %":INPUT RH
80 PRINT"enter the expected cloud cover in %":INPUT CC
90 PRINT"enter trailer length,width & height in ft":INPUT LN,WD,HT
100 PRINT"enter trailer wall heat transfer coeff in Btu/hr/sq.ft/F":INPUT HW
110 PRINT"enter cooling system capacity in Btu/hr":INPUT BTU
120 PRINT"enter total product weight in lbs":INPUT WT
130 PRINT"enter the number of exposed units":INPUT NU
140 PRINT"enter product specific heat in Btu/lb/F":INPUT HC
150 PRINT"enter product respiration rates at 40,60,& 80 F in Btu/ton/day":INPUT     R40,R60,R80
160 PRINT"enter product loading temperature in deg F":INPUT TL
170 PRINT"enter the thermostat set point temperature in deg F":INPUT TSET
180 PRINT"enter avg travel speeds in mph every hr beginning with dawn"
190 FOR HOUR=1 TO 24
200 INPUT VEL(HOUR)
210 NEXT HOUR
220 REM: begins distance & length of day calculations
230 IF ABS(LAT1)<66.5 AND ABS(LAT2)<66.5 THEN 250
240 PRINT"forget it! - you're in the artic/antartic circle" : STOP
```

```
250 LAT=(LAT1+LAT2)/2
260 DAY=(DAY+10.5)*360/365
270 P=(COS(LAT1*3.14159/180)+COS(LAT2*3.14159/180))/2
280 Q=SQR(((LONG2-LONG1)*P)^2+(LAT2-LAT1)^2)
290 DIST=1.15*3950*Q*3.14159/180
300 IF Q>0 THEN 320
310 SV=1 : CV=0 : GOTO 330
320 SV=(LONG1-LONG2)*P/Q : CV=(LAT2-LAT1)/Q
330 ST=SIN(23.45*3.14159/180) : CT=COS(23.45*3.14159/180)
340 SD=SIN(DAY*3.14159/180) : CD=COS(DAY*3.14159/180)
350 SL=SIN(LAT*3.14159/180) : CL=COS(LAT*3.14159/180)
360 IF SD=0 THEN SD=.000001
370 ZEN=ATN(-CT*CD/SD)*180/3.14159
380 IF SD>0 THEN ZEN=ZEN+180
390 P=SQR(CL^2-(ST*CD)^2) : Q=ST*SL*CD
400 IF Q=0 THEN Q=.000001
410 R=ATN(P/Q)*180/3.14159
420 IF R<0 THEN R=R+180
430 DAWN=ZEN-R : DUSK=ZEN+R
440 LDAY=(DUSK-DAWN)/15
450 REM: min/max temps, RH effects, heat transf areas, resp coeffs
460 TMIN=(TL1+TL2)/2 : TMAX=(TH1+TH2)/2
470 IF BTU>0 THEN BTU=BTU/(1+.25*RH/100)
480 AO=2*(LN*WD+LN*HT+WD*HT)
490 AL=6*(LN*WD*HT)^(2/3)*NU^(1/3)
500 C1=6*R40-8*R60+3*R80 : C2=7*R40-12*R60+5*R80 : C3=2*R40-4*R60+2*R80
```

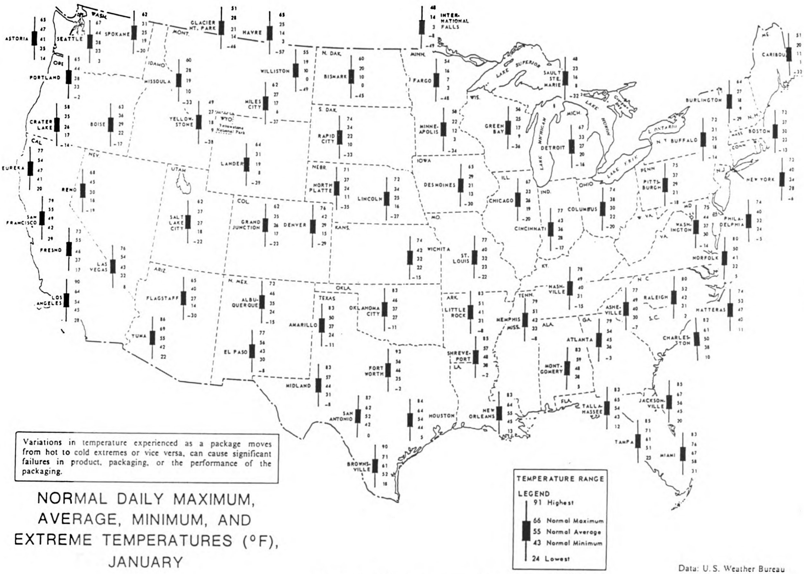
```
510 DTRV=0

520 PRINT"distance = ";DIST;" miles"

530 PRINT "available daylight = ";LDAY;" hrs"

540 PRINT"expected min and max air temperatures are ";TMIN;"F and ";TMAX;"F"

550 PRINT

560 PRINT"hrs past dawn   disttrav   airtemp   skintemp   loadtemp    % duty"

570 PRINT USING"   ######.#";0,0,TMIN,TMIN,TL,0

580 REM: begins time study; calculates every 4 min, prints every hr

590 FOR HOUR=1 TO 24

600 DTRV=DTRV+VEL(HOUR)

610 H1=DAWN+(HOUR-1)*15+1  : H2=H1+14

620 FOR H=H1 TO H2

630 SH=SIN(H*3.14159/180) : CH=COS(H*3.14159/180)

640 SA=(CT*CL*SH-ST*SL)*CD-CL*CH*SD

650 REM: solar insolation part; 10% diffuse, 50% absorption

660 INSOL=0 : IF SA>0 THEN INSOL=442*.81^(1/SA)*(1.1-CC/100)*.5

670 TOP=INSOL*SA*LN*WD

680 FF=(CT*(CH*SV-SL*SH*CV)-ST*CL*CV)*CD+(SH*SV+SL*CH*CV)*SD

690 FRONT=0 : IF FF>0 THEN FRONT=INSOL*FF*WD*HT

700 FR=(CT*(CH*CV+SL*SH*SV)+ST*CL*SV)*CD+(SH*CV-SL*CH*SV)*SD

710 RIGHT=0 : IF FR>0 THEN RIGHT=INSOL*FR*LN*HT

720 FB=-FF

730 BACK=0 : IF FB>0 THEN BACK=INSOL*FB*WD*HT

740 FL=-FR

750 LEFT=0 : IF FL>0 THEN LEFT=INSOL*FL*LN*HT

760 SOLAR = TOP+FRONT+RIGHT+BACK+LEFT
```

```
770 REM: outside air temp, heat transf coeffs, respiration
780 TE=TMIN+(TMAX-TMIN)*(H-DAWN)/(ZEN-DAWN)
790 IF H>ZEN THEN TE=TMAX-(TMAX-TMIN)*(H-ZEN)/(DAWN+360-ZEN)
800 HI=1.15
810 HO=1.27+9*(VEL(HOUR)/65)^.8
820 HEQ=1/(1/HO+1/HW+1/HI)
830 F=HEQ/HO
840 RSP=(C1-C2*(TL/40)+C3*(TL/40)^2)/2000/24
850 REM: begins by assuming chiller/heater not on
860 TA=(F*SOLAR+HEQ*AO*TE+HI*AL*TL)/(HEQ*AO+HI*AL)
870 TS=F*TA+(1-F)*(TE+SOLAR/(HO*AO))
880 DUTY=0 : IF BTU=0 THEN 1000
890 IF BTU>0 AND TA<TSET THEN 1000
900 IF BTU<0 AND TA>TSET THEN 1000
910 REM: if chiller/heater on, assumes capacity adequate & checks
920 TA=TSET
930 TS=F*TA+(1-F)*(TE+SOLAR/(HO*AO))
940 Q=HI*AL*(TL-TA)+F*SOLAR+HEQ*AO*(TE-TA)
950 DUTY=100*Q/BTU : IF DUTY<100 THEN 1000
960 REM: if chiller/heater capacity inadequate, recalculates
970 TA=(F*SOLAR+HEQ*AO*TE+HI*AL*TL-BTU)/(HEQ*AO+HI*AL)
980 TS=F*TA+(1-F)*(TE+SOLAR/(HO*AO))
990 DUTY=100
1000 K1=(RSP*WT)/(HI*AL) : K2=(HI*AL)/(WT*HC)
1010 TL=TA+K1+(TL-TA-K1)*EXP(-K2/15)
1020 NEXT H
```

```
1030 PRINT USING"    ######.#";HOUR,DTRV,TE,TS,TL,DUTY
1040 NEXT HOUR
1050 END
```

APPENDIX D

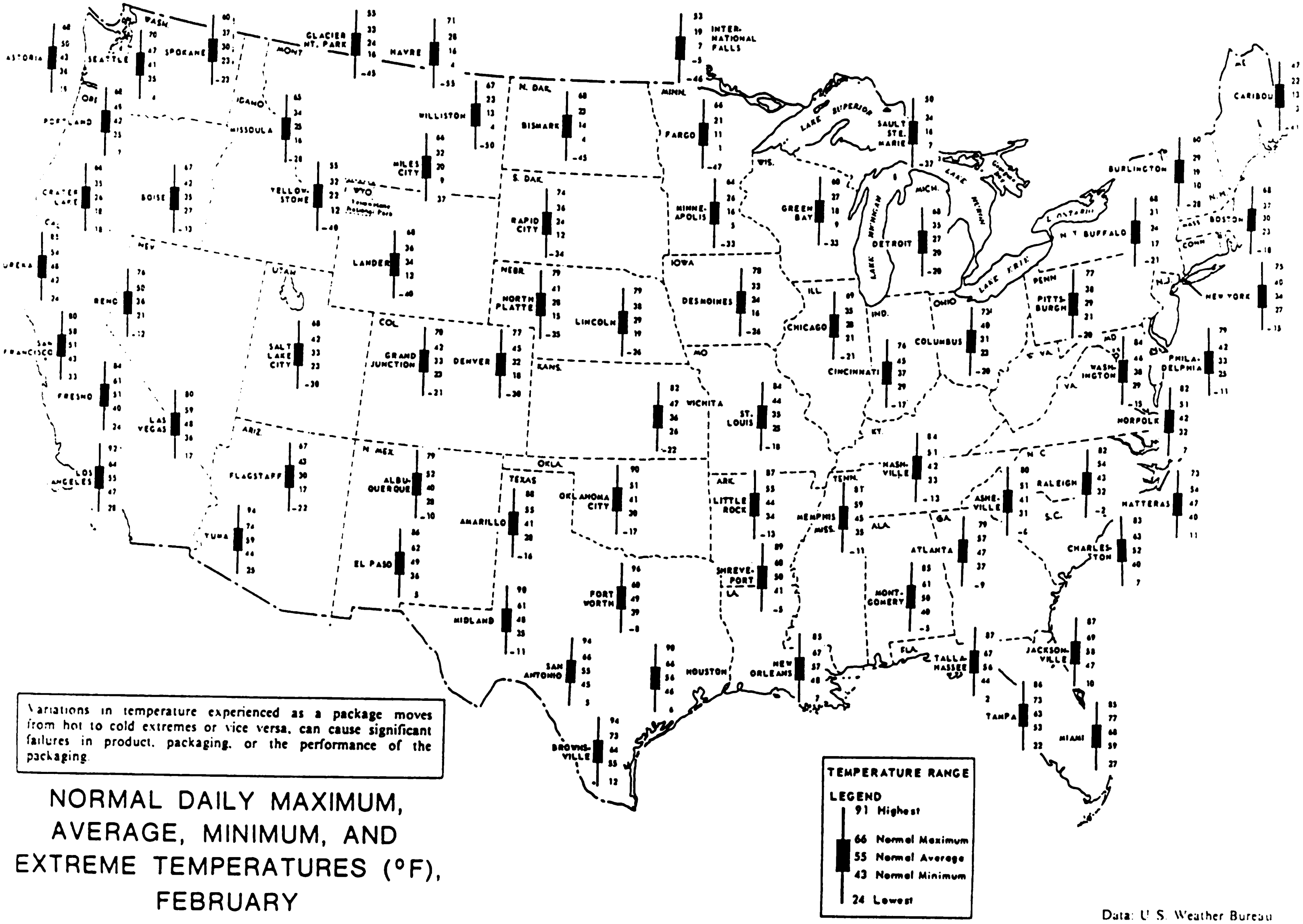
The twelve maps in this appendix show the high and low temperatures of various major cities in the United States throughout the year. They are reproduced from the book "Packaging for the Small Parcel Environment", 1976, with permission from the United Parcel Service.



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), JANUARY

Figure 10. Weather Map of January

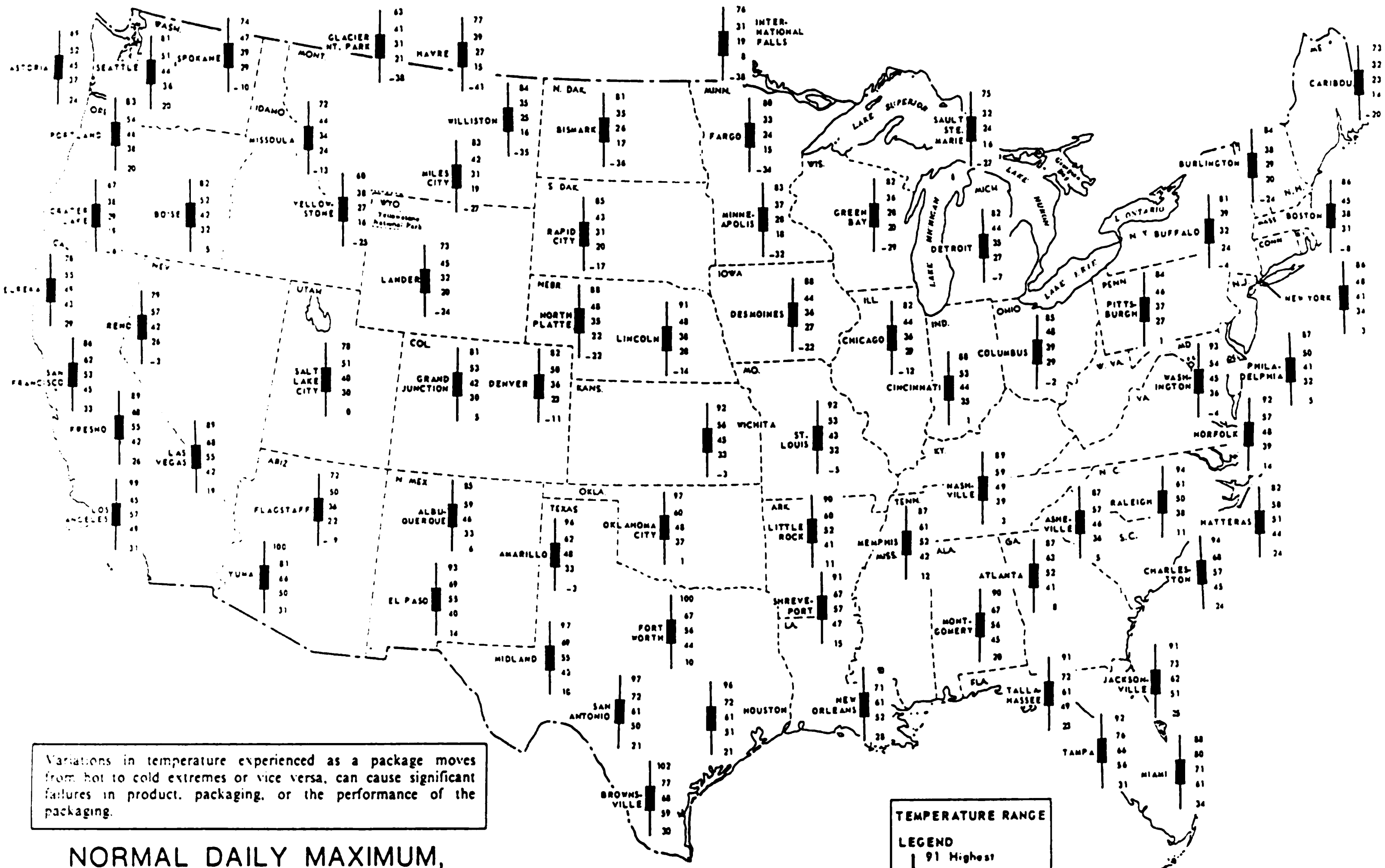


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), FEBRUARY

Data: U.S. Weather Bureau

Figure 11. Weather Map of February

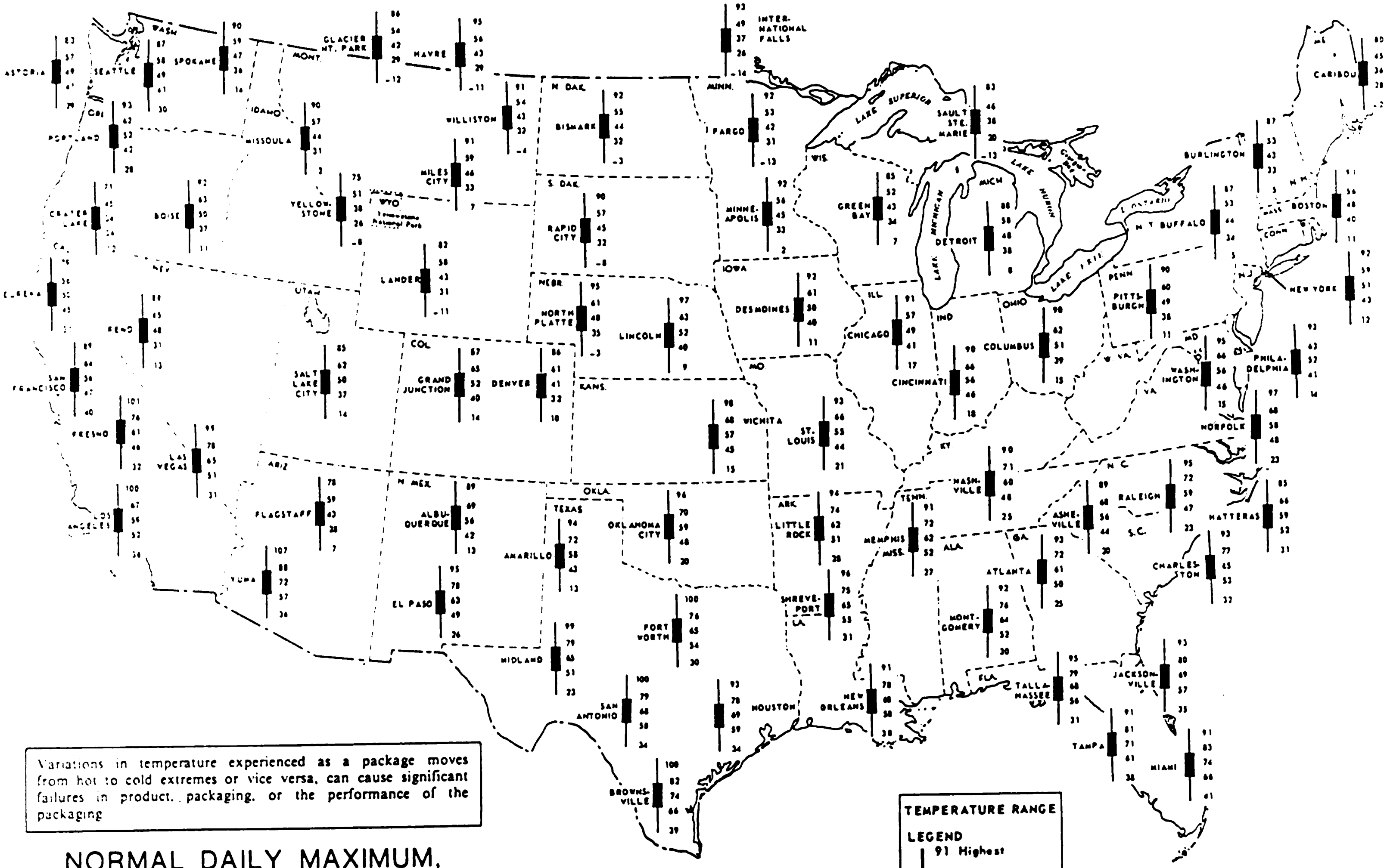


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), MARCH

Data: U.S. Weather Bureau

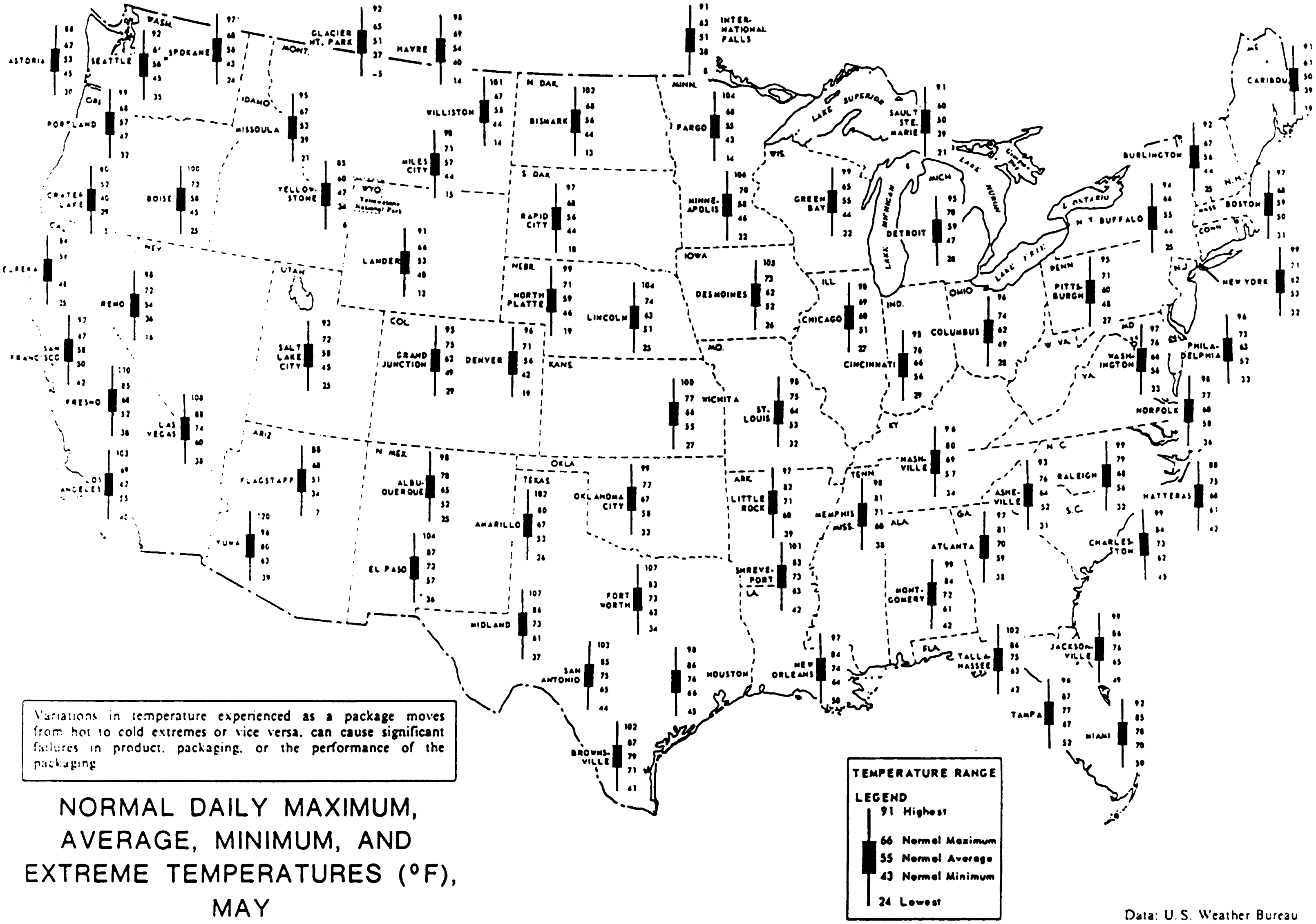
Figure 12. Weather Map of March



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), APRIL

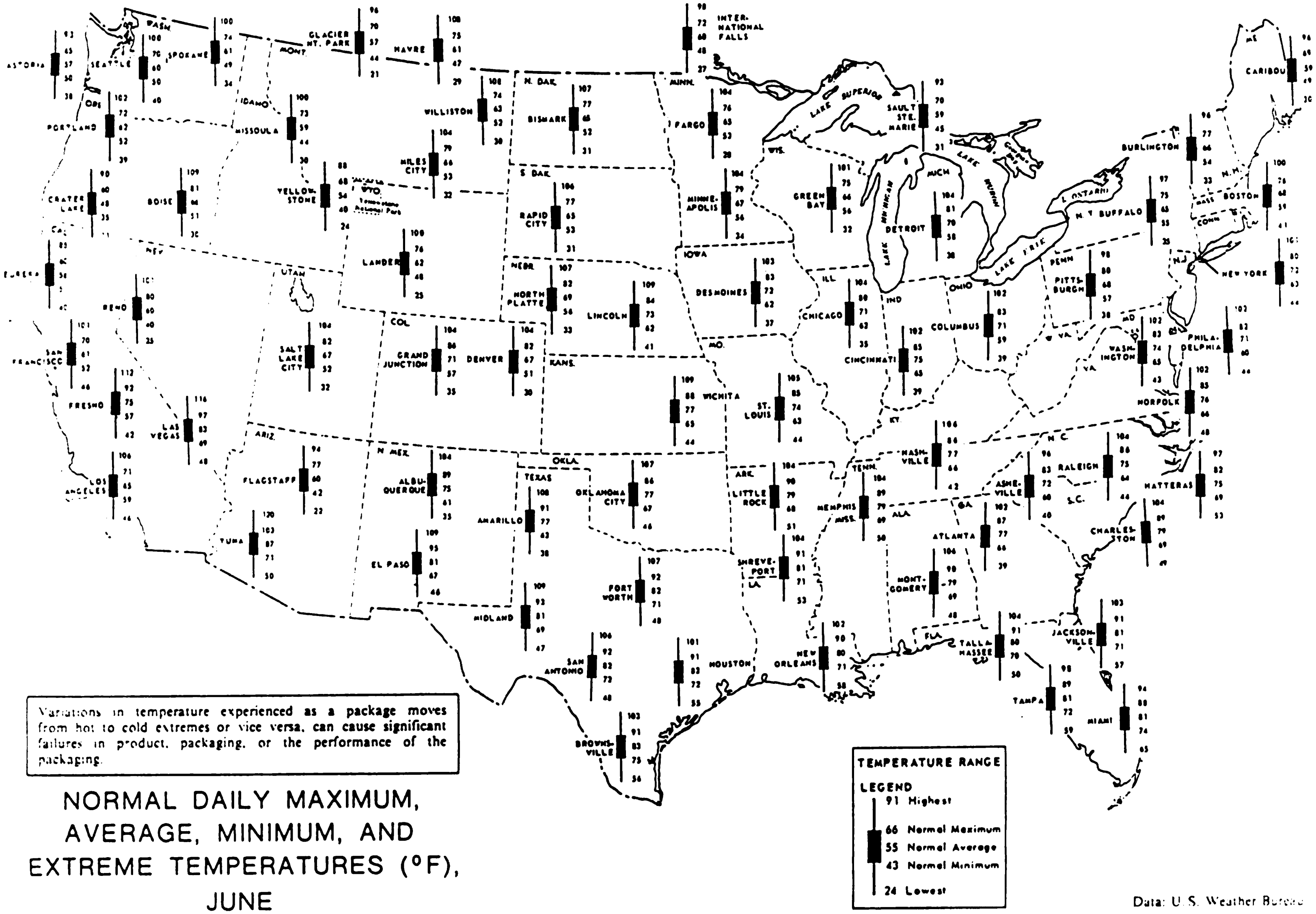
Figure 13. Weather Map of April



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), MAY

Figure 14. Weather Map of May

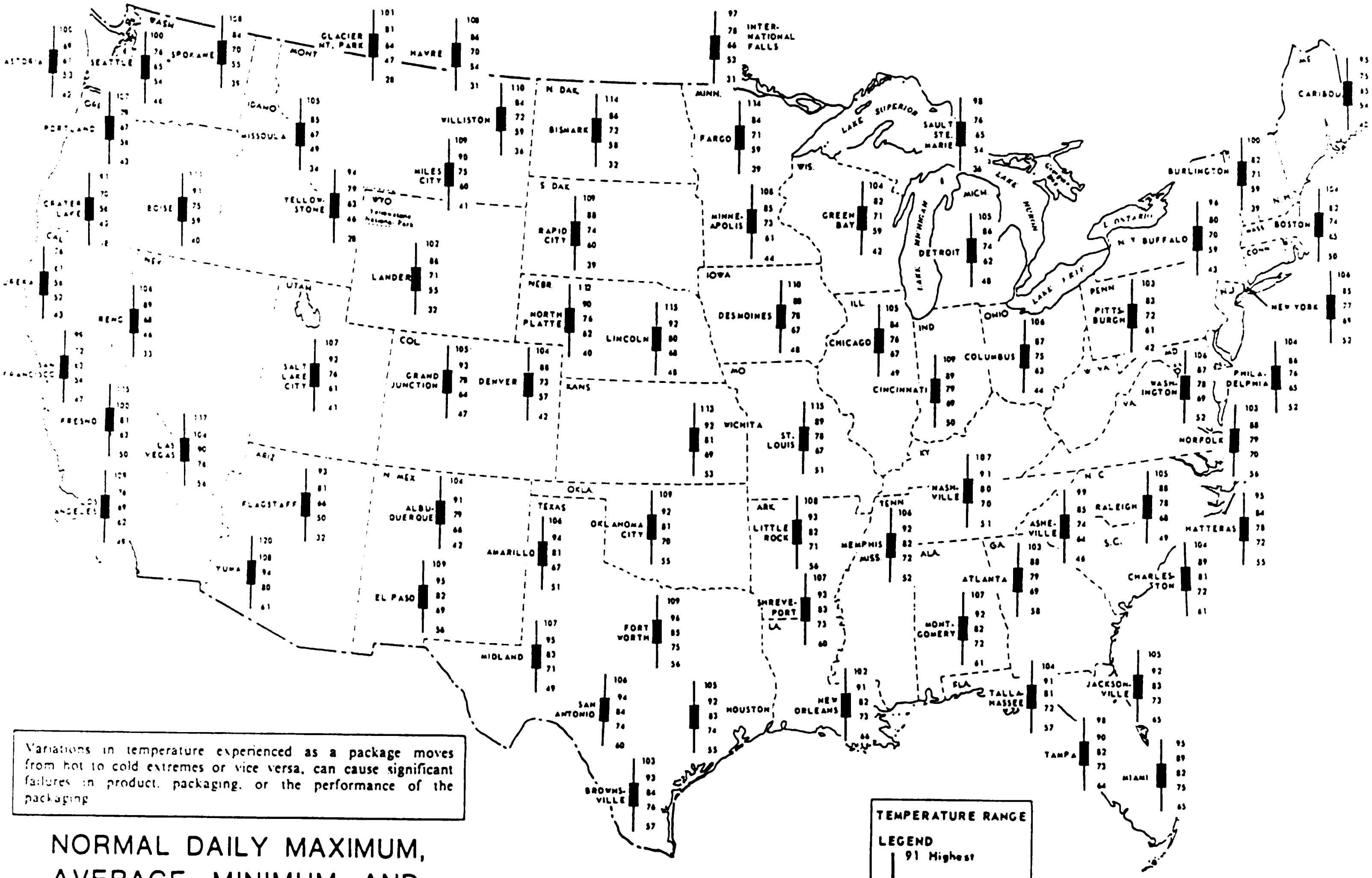


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), JUNE

Data: U.S. Weather Bureau

15. Weather Map of June

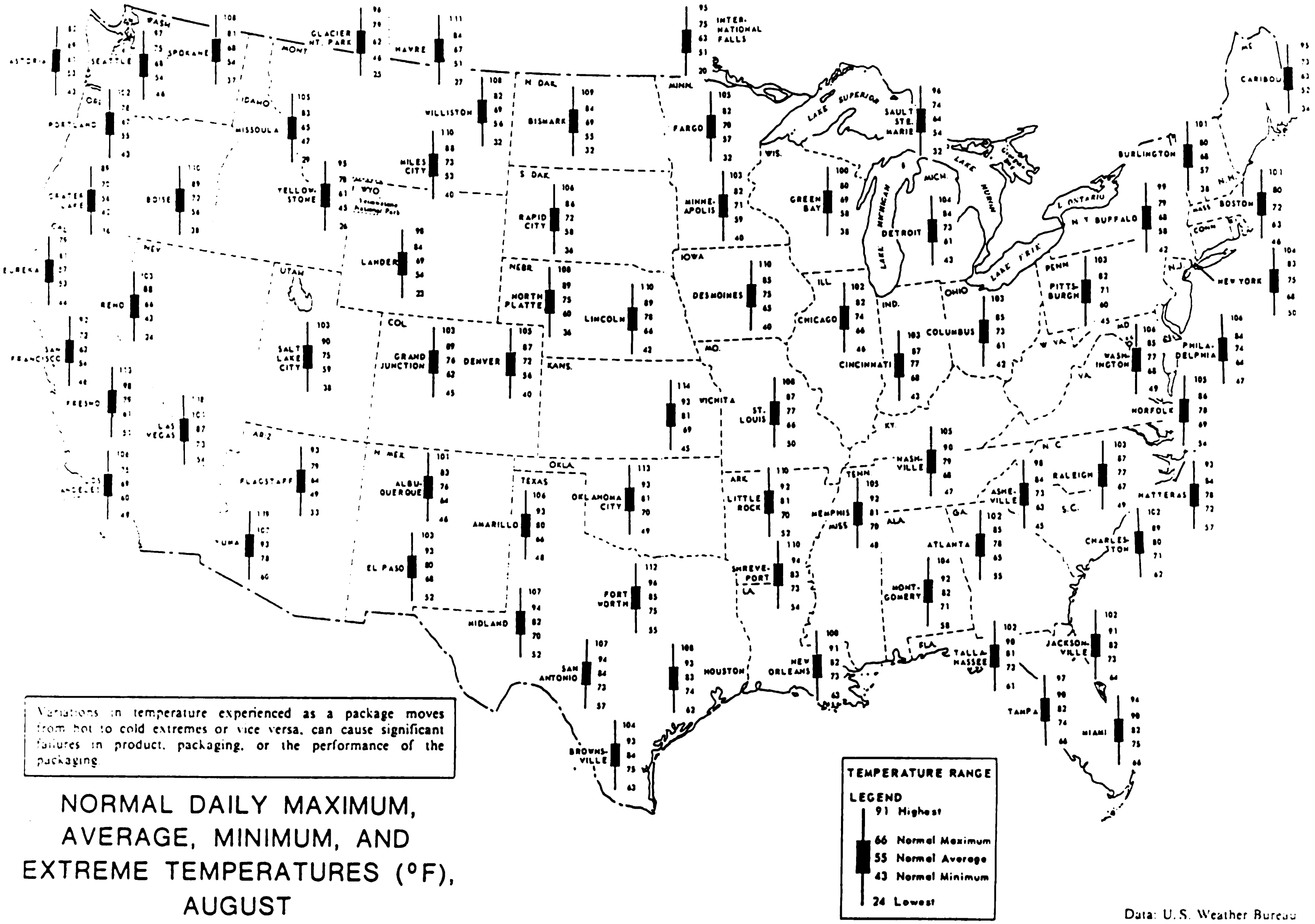


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), JULY

Data: U.S. Weather Bureau

Figure 16. Weather Map of July

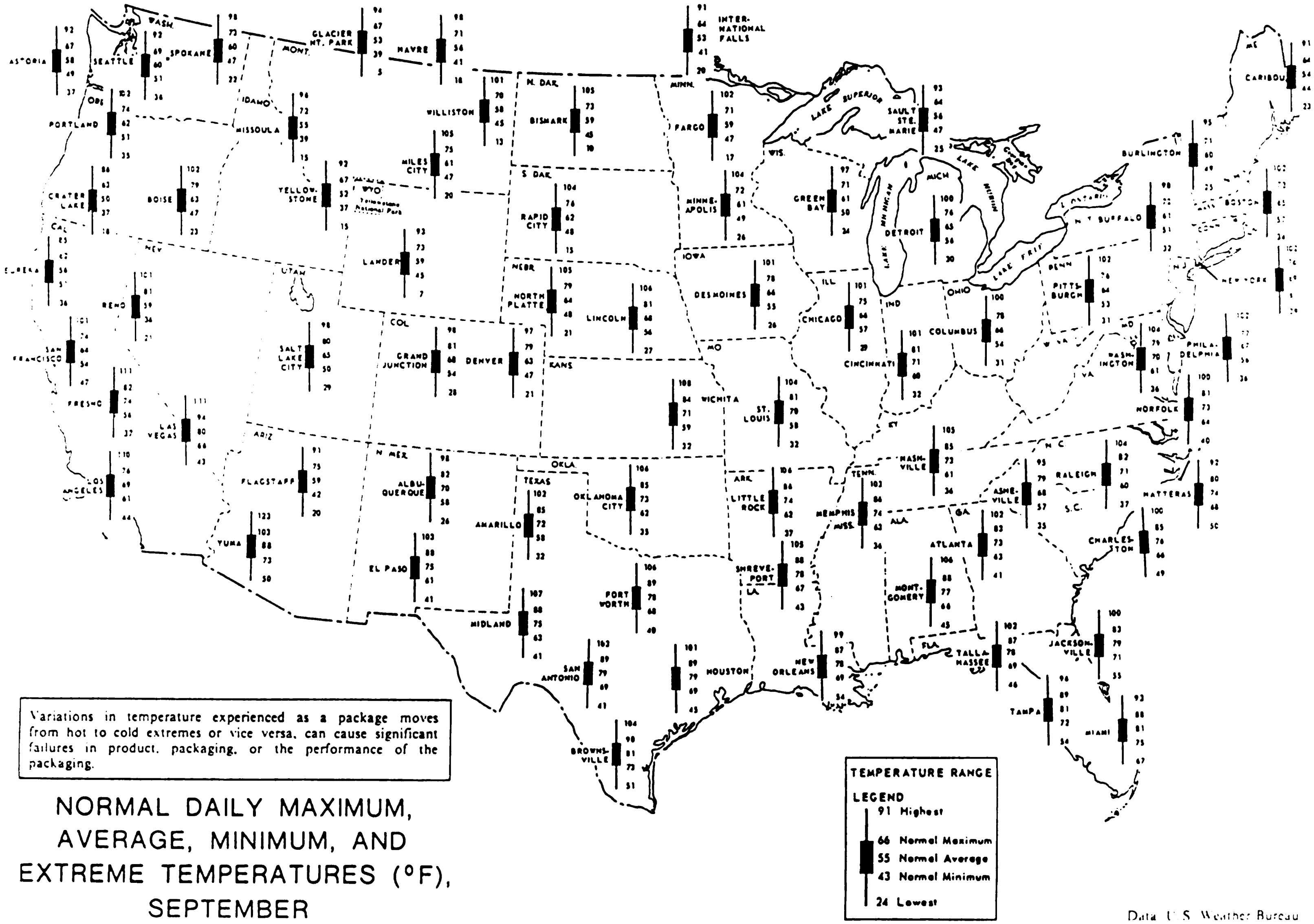


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), AUGUST

Data: U.S. Weather Bureau

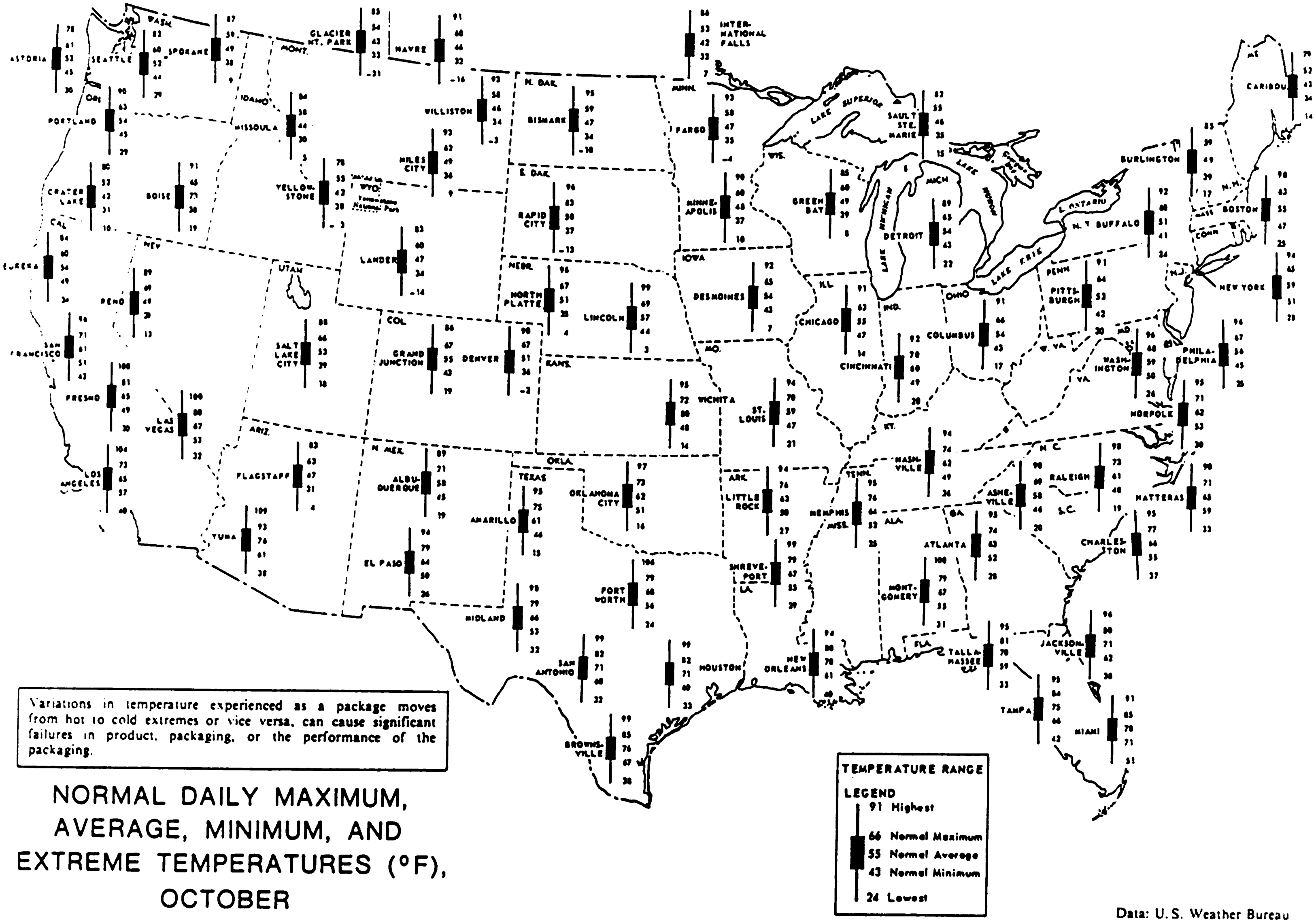
Figure 17. Weather Map of August



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), SEPTEMBER

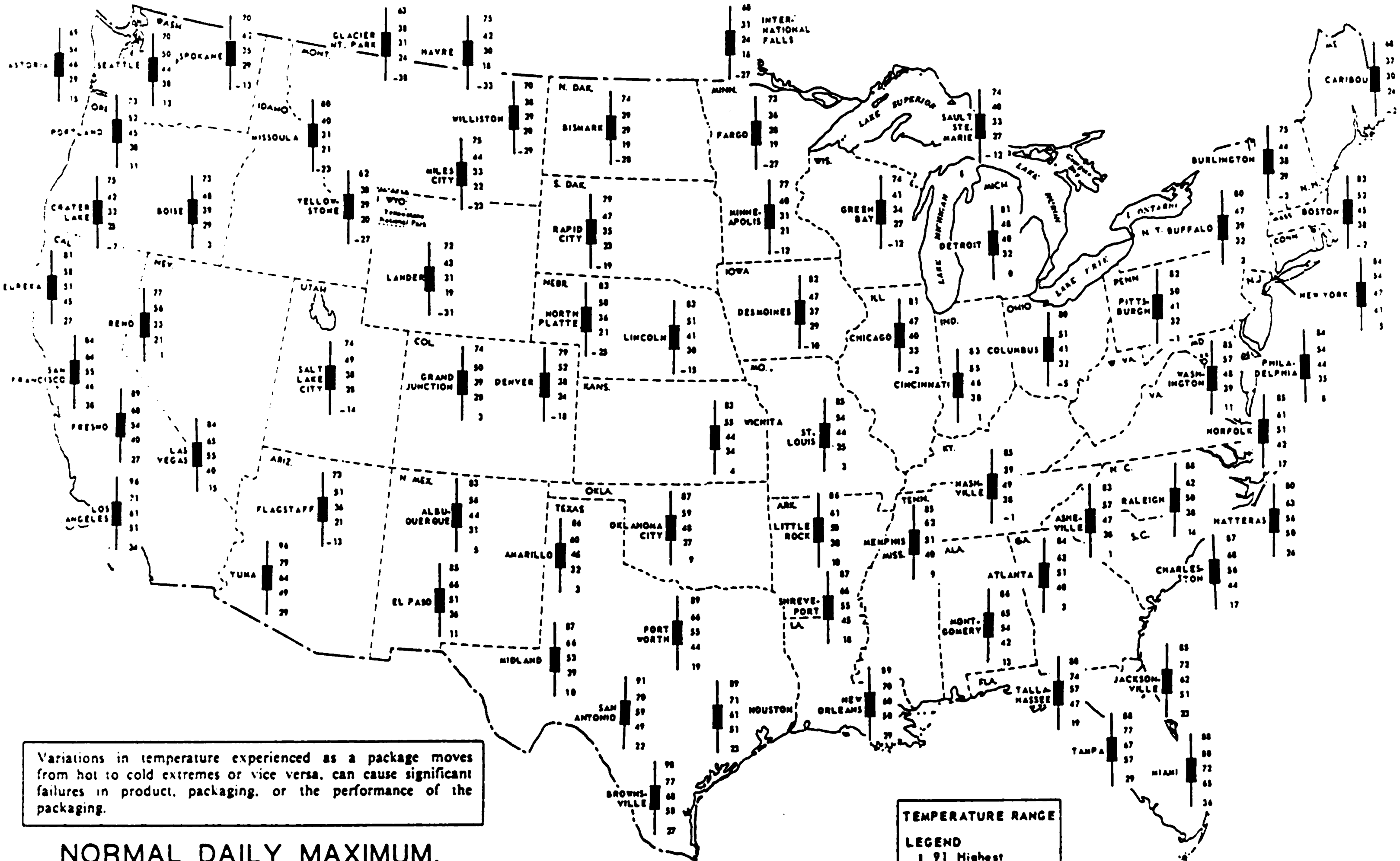
Figure 18. Weather Map of September



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), OCTOBER

Figure 19. Weather Map of October

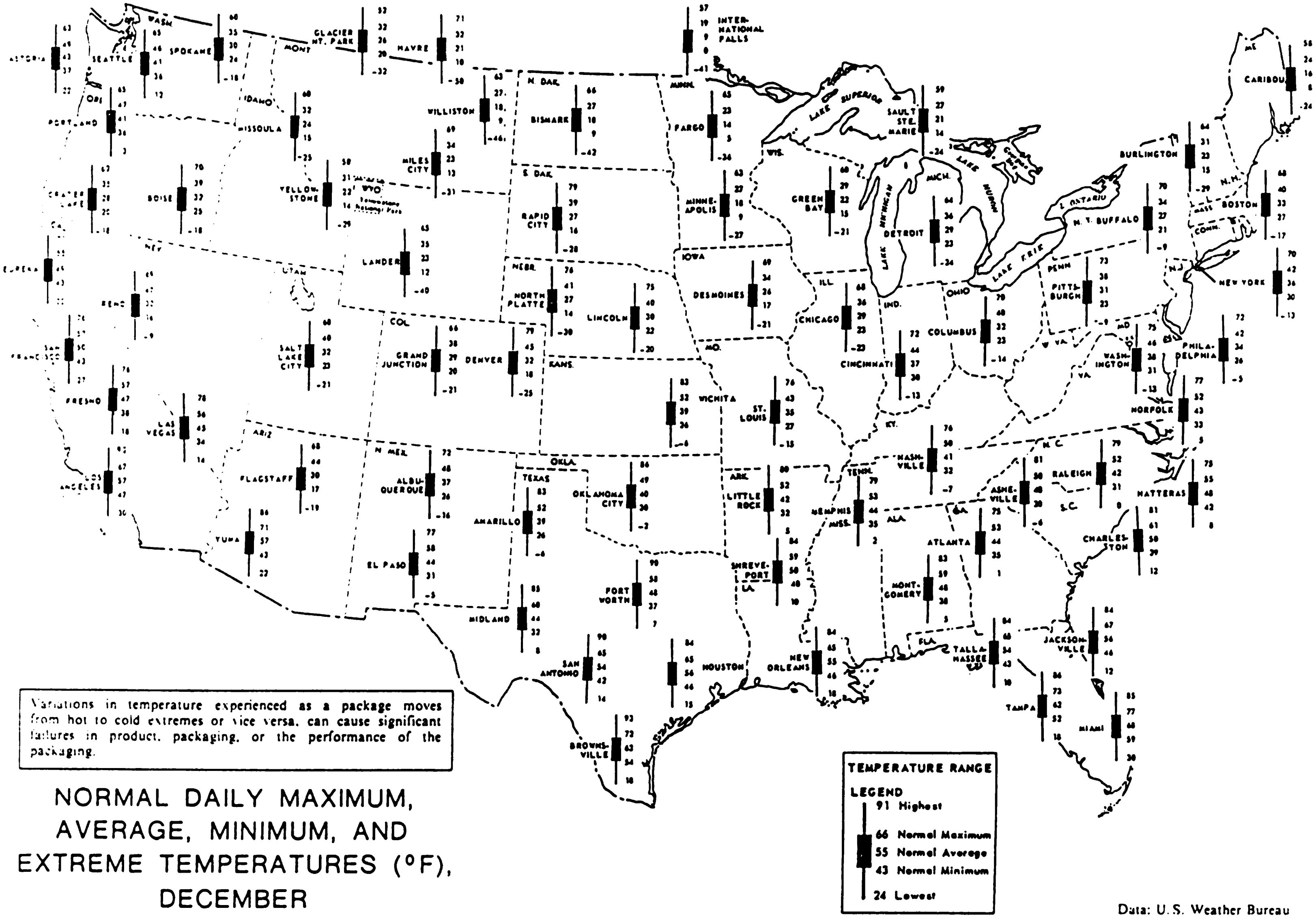


Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), NOVEMBER

Data: U.S. Weather Bureau

Figure 20. Weather Map of November



Variations in temperature experienced as a package moves from hot to cold extremes or vice versa, can cause significant failures in product, packaging, or the performance of the packaging.

NORMAL DAILY MAXIMUM, AVERAGE, MINIMUM, AND EXTREME TEMPERATURES (°F), DECEMBER

Figure 21. Weather Map of December

LIST OF REFERENCES

Burgess, G., A Model for Temperature Variations Inside Trailers and Railcars, Consortium of Distribution Packaging Report, Michigan State University, 1992.

Macro, A., Software Engineering: Concepts and Management, Prentice Hall, 1990.

Mayrhauser, A., Software Engineering: Methods and Management, Academic Press, 1990.

MicroSoft Software Development Kit Reference 3.0, Microsoft, 1990

Microsoft C 6.0 Reference, Microsoft, 1990

Packaging for the Small Parcel Environment, United Parcel Service, 1976.

Petzold, C., Guide to Windows Applications Development, Microsoft Press, 1991.

Pressman, R., Software Engineering - A Beginner's Guide, McGraw-Hill Book Company, 1988.

Smith, D. & Wood, K., Engineering Quality Software - A Review of Current Practices Standards and Guidelines including New Methods and Development Tools, Second Edition, Elsevier Applied Science, 1989.

Sneed, H., Software Engineering Management, Halsted Press, 1989.