Problem: The day known as “Black Friday” is that date which follows Thanksgiving in the United States. It is often viewed as the start of the Christmas shopping season. Retailers go out of their way to have huge sales in an effort to encourage most Americans that have the day off from work to spend money.

Write a program that will allow the user to enter the prices of up to 25 items that they would like to purchase on Black Friday. Next, the user will enter the number of items that they would like to purchase and the amount of money that have available to purchase the products on their list.

Determine whether:

1. All unique combinations of the desired number of items can be purchased;
2. The first desired number of items listed can be purchased;
3. There is at least one combination of the desired number of items that can be purchased.

Example Execution #1:

Enter item price values: 10 5 15 20 30 25 -1
Enter the quantity of desired items: 3
Enter available funds: 100

All unique 3 combination of items can be purchased.

Example Execution #2:

Enter item price values: 45 10 5 15 20 25 30 -1
Enter the quantity of desired items: 3
Enter available funds: 50

There are at least 3 items that can be purchased.

Example Execution #3:

Enter item price values: 10 30 20 50 80 40 110 75 5 -1
Enter the quantity of desired items: 3
Enter available funds: 65

The first 3 items can be purchased.

Example Execution #4:

Enter item price values: 40 50 60 70 40 50 60 30 -1
Enter the quantity of desired items: 3
Enter available funds: 100

No purchases of at least 3 items can be made with the funds indicated.

Academic Integrity Reminder:

- Please review the policies of the course as they relate to academic integrity. The assignment you submit should be your own original work. You are to be consulting only course staff regarding your specific algorithm for assistance. Collaboration is not permitted on individual homework assignments.

All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!
Example Execution #5 (input validation requirements demonstrated):

Enter item price values: 10 20 30 45 35 25 15 -1
Enter the quantity of desired items: 8

Error! Quantity may not exceed the number of items available!!

Enter the quantity of desired items: 7
Enter available funds: -100

Error! Available funds must be non-negative!!

Enter available funds: 200

All unique 7 combination of items can be purchased.

Example Execution #6 (no -1 needed when 25 items are entered):

Enter item price values: 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7
Enter the quantity of desired items: 5
Enter available funds: 20

The first 5 items can be purchased.

Additional Requirements:

1. Add the homework (head_hw) assignment header file to the top of your program. A description of your program will need to be included in the assignment header.
   ○ This particular header can be added to your file by entering hhw while in command mode in vi.

2. Each example execution represents a single test of your program. The six examples provided imply that the program was run six different times.
   ○ Your program is expected to accept input, validate input, and produce output in the same manner demonstrated in the example executions.
   ○ The user will always enter a minimum of one positive item price. All item prices will be positive integers.
   ○ Be sure to test your program with 24 integers and the -1 value as the 25th.

3. For this assignment you will be required to implement the user-defined functions (from chapter 4). Failing to follow course standards as they relate to good user-defined function use will result in a zero for this assignment.
   ○ Each single task must be factored into a unique user-defined function.
   ○ The main function is expected to declare variables and call a majority of the user-defined functions in the program.

4. Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first EIGHT chapters of the book, notes, and lectures to be acceptable for use.
   ○ Use of the course notes packet or official C programming text of the class are both acceptable sources of code for the sorting and searching algorithms. You must document your sources in the relevant function headers.

5. A program MUST compile to be considered for partial credit. The submission script will reject the submission of any file that does not successfully compile on the guru server.

Course Programming and Documentation Standards Reminders:

- Make good use of symbolic/defined constants (example: array maximum size).
- The use of { and } for the body of relevant selection and repetition constructs is required.
- All code found within the body of a selection or repetition construct should be indented two additional spaces.
- See problem 5-7 on page 258 of your C programming text for an example of proper indenting within a switch construct.
- Control-forcing statements such as exit, break, continue, and the use of multiple return statements in a single function are not acceptable practices according to course standards.
Course Programming and Documentation Standards Reminders (continued):

- Use the course function header (`head_fx` vi shortcut `hfx` while in command mode) for every user-defined function in your program.
  - List and comment **all parameters** to a function, one per line, in the course function header.
  - **All function declarations** will appear in the global declaration section of your program.
  - **The user-defined function definitions will appear in your program after the main function.**

- Remove any diagnostic print statements from your code, even if they are commented out (inactive), unless you believe your program to be logically incorrect or incomplete as a way to demonstrate to your lab instructor how much of the problem you were able to solve and attempted to implement.

**When you submit...** only the final attempt of a submission is kept for grading. All other submissions are over-written and cannot be recovered. You may make multiple submissions but only the last attempt is retained and graded.

- Verify in the confirmation e-mail sent to you by the course that you have submitted the correct file, to the correct assignment (`hw07`), and to the correct section.
- Leave time prior to the due date to seek assistance should you experience difficulties completing or submitting this assignment.
- All attempts to submit via a method other than through the guru server as set up during the first week of the semester will be denied consideration.

**Assignment deadlines...** are firm and the electronic submission will disable promptly as advertised. We can only grade what you submit as expected prior to the assignment deadline.