Homework 2: Array problem II

Please do this homework at least three problems, but problem 1 , 2, 3must be selected. Do more as a bonus.

(任選四題,但第一二三題一定要作, 多作當成加分題)

第一題:

How to pack as much value with a weight constraint W? Selection of n=4 items, capacity of knapsack is W=8(total weight)

The following table lists the value and the corresponding weight for each item:

|  |  |  |
| --- | --- | --- |
| Item i | Value | Weight |
| 1 | 15 | 1 |
| 2 | 10 | 5 |
| 3 | 9 | 3 |
| 4 | 5 | 4 |

Please write a program to select the items with maximum value under the constraint of weight W=8.

Please print the maximum value and list the selected items. **(5%)**

第二題: Use “vector” only

Given a positive integer *n*, print out the position of all 1’s in its binary representation. The position of the least significant bit is 0.

Example:

The positions of 1’s in the binary representation of 13 are 0, 2,3.

第三題:

Write a function that will merge(合併) the contents of two sorted(ascending order) arrays of type double values, storing the result in an array output parameter (still in ascending order). The function should not assume that both its input parameter arrays are the same length but can assume that one array does not contain two copies of the same value. The result array should also contain no duplicate values.

Example:

First array

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -10.5 | -1.8 | 3.5 | 6.3 | 7.2 |

4

3

2

1

0

|  |  |  |
| --- | --- | --- |
| -1.8 | 3.1 | 6.3 |

Second array

Result array

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -10.5 | -1.8 | 3.1 | 3.5 | 6.3 | 7.2 |

Hint:

When one of the input arrays has been exhausted, do not forget to copy the remaining data in the other array into the result array.

Test your function n with cases in which

1. The first array is exhausted first,
2. The second array is exhausted first,
3. The two array are exhausted at the same time(i.e. they end with the same value).

Remember that the arrays input to this function must already be sorted.

第四題:

Write a program that reads in a set of positive integers, representing test scores for a class, and outputs how many times a particular number appears in the list. You may assume that the data set has at most 100 numbers and -999 marks the end of the input data. The numbers must be output in increasing order.

For example, the data set:

55 80 78 92 95 55 78 53 92 65 78 95 85 92 85 95 95 -999

The output is:

Test Score Count

53 1

55 2

65 1

78 3

80 1

85 2

92 3

95 4

第五題: (Airplane Seating Assignment)

Write a program that can be used to assign seats for a commercial airplane.

The airplane has 15 rows, with 9 seats in each row. Rows 1 and 2 are first

class, rows 3 through 8 are business class, and rows 9 through 15 are

economy class. Your program must prompt the user to enter the following

information:

1. Ticket type(first class, business class, or economy class)
2. Desired seat

Output the seating plane in the following form:

A B C D E F G H I

Row 1 x \* x \* x \* \* x x

Row 2 \* x \* x \* \* \* \* x

Row 3 x \* \* \* \* \* x x x

Row 4 x \* \* \* \* x x x x

Row 5 \* x x \* \* \* \* x x

Row 6 x x x x x \* \* x x

Row 7 x \* \* \* x \* \* \* x

Row 8 x x x \* \* \* \* x x

Row 9 x x x x \* \* x x x

Row 10 x x \* \* x x x x x

Row 11 \* \* \* \* \* \* \* x x

Row 12 x x \* \* \* \* x x x

Row 13 x \* \* \* \* \* \* x \*

Row 14 \* \* \* \* \* \* \* \* x

Row 15 \* x \* \* \* \* \* \* \*

Where, \* indicates that the seat is available; x indicates that the seat is occupied. Try to make this a menu-driven program; show the user’s choices and allow the user to make the appropriate choices.

第六題: Given a set of points, find the two points that are nearest to each

other. This is called the closest-pair problem.

Input:

The point set: (-1, 3), (-1, -1), (1, 1), (2, 0.5), (2, -1), (3, 3), (4, 2), (4, -0.5)

Output:

The closest two points are (1, 1), (2, 0.5)

第七題:

Write an interactive program that plays tic-tac-toe.(井字遊戲)

Represent the board 3 × 3 character array.

Initialize the array to blanks and ask each player in turn to input a position. The first player’s is marked on the board with an O, and the second player’s position is marked with X. Continue the process until a player wins or the game is a draw (平手). To win, a player must have three marks in a row, in a column, or on a diagonal. A draw occurs when the board is full and no one has won.

Each player’s position should be input as indexes into the tic-tac-toe board:

that is, a row number, a space, and a column number. Make the program more user friendly as you can do.

After each game, print out a diagram of the board showing the ending position. Keep a count if the number of games each player has won and the number of the draws. Before the beginning of each game, ask each player if he or she wishes to continue. If either player wishes to quit, print out the statistics and stop.

請同學儘己所能發揮 甚至可以4 × 4 or 5 × 5 …

也可以作成one player和 電腦 玩

沒有經驗的同學只要將基本的部份作出即可

# 補充

# CSTools C Library 之random number function :

1.#include<stdlib.h>

void srand(unsigned int seed);

2.#include<stdlib.h>

int rand(void) ; 0 ≦ rand() ≦ 32767

若在呼叫rand()之前沒有呼叫srand(x); where x is 正整數

則系統會自動先呼叫srand(1);再叫rand();

第七題: Divisible group sums (bonus 加分題)

Given an array of *n* integers, an integer *k*, , and an integer *d*, . For all groups of *k* integers chosen from the array, determine how many of these chosen groups have a sum, which is divisible by *d*.

Input

The first line of input file contains an integer number *M* specifying the number of test cases. Each test case consists of a single line. In the beginning of test case are 3 decimal numbers indicating (*n, k, d*). Following the 3 decimal numbers is the array of *n* integers. All numbers in the same line are interleaved by a space.

Output

The output data file contains *M* lines, each of which is the number of groups for the corresponding input case. For the *i*-th case, you should output Case *i*: followed by a space and the number of groups in line *i*.

Sample Input

3

7 2 5 1 3 5 6 7 8 9

7 5 3 1 3 5 6 7 8 9

7 6 7 1 3 5 6 7 8 9

Sample Output for the Sample Input

Case 1: 4

Case 2: 7

Case 3: 0

All inputs are from stdin(keyboard) and outputs to stdout(screen).