**81.registerAccountNumbers**

Read the question carefully and follow the input and output format.  
  
Given an array in which the elements are in xxxyy format, where first xxx digits represent the Branch code and the yy represents the account  
id.  Find out the No of accounts in the given branch code  
  
Input and Output Format :  
The first input n corresponds to the size of the array, the next n lines correspond to the elements of the array and the last line of the input corresponds to the branch code.  
Output corresponds to the number of accounts in the given branch code  
If the given branch code is not available, print 0.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid account Number" when there is any negative number available in the input array and terminate the program  
3) Print "Invalid branch code" when branch code is negative number and terminate the program  
  
Include a function named registerAccountNumbers (int size, int account\_numbers[], int branch\_code) that returns the no of accounts  
  
**Sample Input 1 :**  
6  
12345  
12370  
12324  
13355  
13333  
14575  
123  
  
**Sample Output 1 :**  
3  
  
**Sample Input 2 :**  
-6  
  
**Sample Output 2:**  
Invalid array size

#include<stdio.h>

#include<stdlib.h>

int registerAccountNumbers (int size, int account\_numbers[], int branch\_code);

int main(){

int n=0,no\_of\_acc=0,input[30],i,flag=0,bcode=0;

scanf("%d",&n);

if(n<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0)

flag=1;

}

if(flag==1){

printf("Invalid account number");

getchar();

getchar();

exit(0);

}

scanf("%d",&bcode);

if(bcode<0){

printf("Invalid branch code");

getchar();

getchar();

exit(0);

}

no\_of\_acc=registerAccountNumbers(n,input,bcode);

printf("%d",no\_of\_acc);

getchar();

getchar();

return 0;

}

int registerAccountNumbers(int size, int account\_numbers[], int branch\_code){

int count=0,i,rem,temp=0,k=1,bcode=0,x=1;

for(i=0;i<size;i++){

temp=account\_numbers[i];

bcode=temp/100;

if(bcode==branch\_code)

count++;

}

return count;

}

**82.clearedStage1**

Read the question carefully and follow the input and output format.  
  
Given an integer array. The first index represents the Student id, Second index represents C-programming marks and the third index Represents SQL marks. Write a program to find the Ids of students who have cleared both C-programming and SQL.  
  
Note :(1) The Pass Marks is >=70  
  
**Input and Output Format :**  
  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements. Output consist of an integer array.  
  
1) Print "Invalid array size" when size of the array is negative and terminate the program.  
2) Print "Invalid input" when there is any negative number available in the input array and terminate the program.  
  
Include a function named clearedStage1(int array[], int size) whose return type is void.  
The output array is stored in a global variable named cleared.  
  
**Sample Input 1:**  
9  
1  
25  
75  
3  
75  
80  
2  
75  
75  
  
**Sample Output 1:**  
3  
2  
  
**Sample Input 2:**  
6  
4  
25  
-78  
  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

int registerAccountNumbers (int size, int account\_numbers[], int branch\_code);

int main(){

int n=0,no\_of\_acc=0,input[30],i,flag=0,bcode=0;

scanf("%d",&n);

if(n<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0)

flag=1;

}

if(flag==1){

printf("Invalid account number");

getchar();

getchar();

exit(0);

}

scanf("%d",&bcode);

if(bcode<0){

printf("Invalid branch code");

getchar();

getchar();

exit(0);

}

no\_of\_acc=registerAccountNumbers(n,input,bcode);

printf("%d",no\_of\_acc);

getchar();

getchar();

return 0;

}

int registerAccountNumbers(int size, int account\_numbers[], int branch\_code){

int count=0,i,rem,temp=0,k=1,bcode=0,x=1;

for(i=0;i<size;i++){

temp=account\_numbers[i];

bcode=temp/100;

if(bcode==branch\_code)

count++;

}

return count;

}

**83.findLargest**

Read the question carefully and follow the input and output format.  
  
Write a program to find the largest of the 3 given numbers.  
  
**Input and Output Format :**  
Input consists of 3 integers. Output consist of an integer that is the maximum.  
  
Print "Number too large" when any of given input numbers is greater than 32767 .  
Print "Number too small" when given input is a negative number.  
  
Include a function named findLargest(int num1, int num2, int num3) whose return type is an integer, which is the largest.  
  
**Sample Input 1:**  
2  
3  
4  
  
**Sample Output 1:**  
4  
  
**Sample Input 2:**  
98974  
  
**Sample Output 2:**  
Number too large  
  
**Sample Input 3:**  
-32767  
  
**Sample Output 3:**  
Number too small

**84.sumThreeLargest**

Read the question carefully and follow the input and output format.  
  
Write a program to find the sum of first ,second and third largest element in the given array.  
  
**Input and Output Format:**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements . Output consists of an Integer, the sum.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program.  
2) Print "Invalid input" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named sumThreeLargest(int array[], int n) whose return type is integer  
  
**Sample Input 1:**  
8  
1  
2  
2  
3  
4  
5  
5  
4  
  
**Sample Output 1:**  
12  
  
**Sample Input 2:**  
4  
1  
2  
-3  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

int sumThreeLargest(int array[], int n);

int main(){

int n=0,input[20]={0},i,largest;

scanf("%d",&n);

if(n<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

}

largest = sumThreeLargest(input,n);

printf("%d",largest);

getchar();

getchar();

return 0;

}

int sumThreeLargest(int array[], int n){

int i,j,k=0,temp=0;

for(i=0;i<n;i++){

for(j=i+1;j<n;){

if(array[i]==array[j]){

for(k=j;k<n;k++)

array[k]=array[k+1];

n--;

}

else

j++;

}

}

for(i=0;i<n;i++){

for(j=i+1;j<n;j++){

temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

return array[0]+array[1]+array[2];

}

**85.highestProfitYear**

Read the question carefully and follow the input and output format.  
  
An array holds information of a company profit margin and year. Find out the year in which highest revenue was earned. Assume the first index of array indicates year and the next index indicates the amount of money earned by the company and so on.  
  
**Input and Output Format :**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements. Output consist of an integer, which is the sum  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program.  
2) Print "Invalid input" when there is any negative number available in the input array and terminate the program.  
  
Include a function named highestProfitYear(int revenue[],int size) whose return type is an integer, which is the year.  
  
**Sample Input 1:**  
6  
2012  
10000  
2011  
5000  
2009  
4000  
  
**Sample Output 1:**  
2012  
  
**Sample Input 2:**  
8  
2015  
89745  
-2015  
  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

int highestProfitYear(int revenue[],int size);

int main(){

int n=0,input[20]={0},i,highest;

scanf("%d",&n);

if(n<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

}

highest = highestProfitYear(input,n);

printf("%d",highest);

getchar();

getchar();

return 0;

}

int highestProfitYear(int revenue[],int size){

int i,j,k=0,highest=0,max=0;

for(i=1;i<size;i=i+2){

if(revenue[i]>max){

max=revenue[i];

highest=revenue[i-1];

}

}

return highest;

}

**86.countNoOfConnections**

Read the question carefully and follow the input and output format.  
  
Given an input array, the elements are of format XYYY . where X represents the connection type. YYY represents the connection id.  
  
If X is 2 -> means 2G connection  
If X is 3 -> means 3G connection  
If X is 4 -> means 4G connection  
You need to find the number of type of connections.  
  
Note:  
If a particular connection type (starting with 2 or 3 or 4) is not available represent with zero in the corresponding position.  
  
Include a function named countNoOfConnections(int connection\_list[],int no) that returns the number of types of connections  
  
**Business Rules :**  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program.  
2) Print "Invalid connection" when there is any negative number available in the input array and terminate the program  
  
**Input and Output Format :**  
First line of input corresponds to n, next n lines corresponds to the elements of the array  
Output consists of the the number of type of connections. [1st line of the output corresponds to the number of 2G connections, 2nd line corresponds to the number of 3G connections and 3rd line corresponds to the number of 4G connections]  
  
**Sample Input 1 :**  
5  
2333  
3101  
2102  
4567  
3123  
  
**Sample Output 1:**  
2  
2  
1  
  
**Sample Input 2 :**  
2  
-2234  
  
  
**Sample Output 2:**  
Invalid connection  
  
#include<stdio.h>

#include<stdlib.h>

void countNoOfConnections(int connection\_list[],int no);

int main(){

int n=0,input[20]={0},i;

scanf("%d",&n);

if(n<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0){

printf("Invalid Connection");

getchar();

getchar();

exit(0);

}

}

countNoOfConnections(input,n);

getchar();

getchar();

return 0;

}

void countNoOfConnections(int connection\_list[],int no){

int i,j,k=0,temp=0,x=0,two=0,three=0,four=0,other=1,out[10]={0};

for(i=0;i<no;i++){

temp=connection\_list[i];

x=temp/1000;

if(x==2)

two++;

else if(x==3)

three++;

else if(x==4)

four++;

else

other=0;

}

out[0]=two;

out[1]=three;

out[2]=four;

for(i=0;i<3;i++)

printf("%d\n",out[i]);

}

**87.Sum of positive numbers in Array**

Write a program to find the sum of positive numbers in an array.

Include a function named**addPositives** that accepts 2 arguments and returns an int. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The function returns the sum of the positive numbers in the array.

If the size of the array is negative, print “Invalid Input” and terminate the program..

**Input and Output Format:**

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Assume that the maximum size of the array is 20.

**Sample Input 1:**

5

3

5

-2

6

-6

**Sample Output 1:**

14

**Sample Input 2:**

**-5**

**Sample Output 2:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int addPositives(int array[],int n);

int main(){

int n=0,input[20]={0},i,sum;

scanf("%d",&n);

if(n<0){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++)

scanf("%d",&input[i]);

sum=addPositives(input,n);

printf("%d",sum);

getchar();

getchar();

return 0;

}

int addPositives(int array[],int n){

int i,j,sum=0;

for(i=0;i<n;i++){

if(array[i]>0)

sum=sum+array[i];

}

return sum;

}

**88.Find Index**

Write a program to find the index of a particular number in a given input array.

Include a function named **findIndex** that accepts 3 arguments and returns an int. The first argument is the input array, the second argument is an int that corresponds to the size of the array and the third argument is the element to be searched for. The function returns the corresponding index if the search element is present in the array and returns -1 if the search element is not present in the array.

If the size of the array is negative or if any element in the array is negative, print “Invalid Input” and terminate the program.

**Input and Output Format:**

Input consists of n+2 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array. The last integer corresponds to the element whose count needs to be found.

Output consists of an integer that corresponds to the index of the search element if it is present.

Else, print 'not found'.

Refer sample output for formatting specifications.

Assume that the maximum number of elements in the array is 20 and that all elements in the array are unique.

**Sample Input 1:**

8

2

1

3

8

6

12

10

19

8

**Sample Output 1:**

3

**Sample Input 2:**

8

2

1

3

8

6

12

10

19

80

**Sample Output 2:**

not found

**Sample Input 3:**

-5

**Sample Output 3:**

Invalid Input

**Sample Input 4:**

5

23

2

-200

**Sample Output 4:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int cricketer[20];

int findIndex(int,int[],int);

int main(){

int n=0,index=0,input[30],i,search\_element=0;

scanf("%d",&n);

if(n<0){

printf("Invalid Input");

getchar();

getchar();

exit(0);

}

for(i=0;i<n;i++){

scanf("%d",&input[i]);

if(input[i]<0){

printf("Invalid Input");

getchar();

getchar();

exit(0);

}

}

scanf("%d",&search\_element);

index = findIndex(n,input,search\_element);

if(index==-1)

printf("not found");

else

printf("%d",index);

getchar();

getchar();

return 0;

}

int findIndex( int size,int array[], int search){

int index,i,f=0;

for(i=1 ; i<size ; i=i+2)

{

if(array[i]==search){

f=1;

index = array[i-1];

break;

}

}

if(f==1)

return index;

else

return -1;

}

**89.Store Consequtives**

Write a program to obtain a new array that contains the consequtive values of the given input array. The output array is named as output1.

Include a function named **storeConsequtives** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size of the array .  The output array is stored in a global variable named output1.

If the size of the array is negative or if any of the elements in the array are negative , print “Invalid Input” and terminate the program.

**Input and Output Format:**

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Output consists of an integer array.

Refer sample output for formatting specifications.

Assume that the maximum number of elements in the array is 20.

**Sample Input 1:**

4

2

5

1

4

**Sample Output 1:**

3

6

2

5

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int output1[20];

void storeConsequtives(int numbers[], int size);

int main(){

int size=0,input1[20],i=0;

scanf("%d",&size);

if(size<0){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

for(i=0;i<size;i++){

scanf("%d",&input1[i]);

if(input1[i]<0){

printf("Invalid Input");

getchar();

getchar();

exit(0);

}

}

storeConsequtives(input1,size);

return 0;

}

void storeConsequtives(int numbers[], int size){

int i=0,j=0;

for(i=0;i<size;i++){

output1[i]=numbers[i]+1;

printf("%d\n",output1[i]);

}

getchar();

getchar();

}

**90.findGrade**

Read the question carefully and follow the input and output format.  
  
In a school examination the result of students are published in the form of an array where first index is the student id and the second index is the total marks in mathematics third index is student id and fourth index is total marks in mathematics and so on... Write a method that assigns the student id as the key and the grade in mathematics as the value to the output array based on the following conditions:  
  
If(Marks >=90 ) : 1  
If(Marks >=80 and <90 ) : 2  
If(Marks >=70 and <80) : 3  
If(Marks <70 ) : 0  
  
Hint: Array size is always is even.  
  
**Input and Output Format :**  
  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements. Output consist of an integer array.  
1) Print "Invalid array size" when size of the array is negative and terminate the program.  
2) Print "Invalid input" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named findGrade(int array[], int size) whose return type is void.  
The output array is stored in a global variable named grade.  
  
**Sample Input 1:**  
10  
1  
65  
2  
74  
3  
86  
4  
95  
6  
69  
**Sample Output 1:**  
1  
0  
2  
3  
3  
2  
4  
1  
6  
0  
  
**Sample Input 2:**  
4  
-3  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

int grade[20];

void findGrade(int array[], int size);

int main(){

int size=0,input1[20],i=0;

scanf("%d",&size);

if(size<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<size;i++){

scanf("%d",&input1[i]);

if(input1[i]<0){

printf("Invalid Input");

getchar();

getchar();

exit(0);

}

}

findGrade(input1,size);

return 0;

}

void findGrade(int array[], int size){

int i=0,j=0;

for(i=1;i<size;i=i+2){

if(array[i]>=90){

grade[i-1]=array[i-1];

grade[i]=1;

}

else if(array[i]>=80 && array[i]<90){

grade[i-1]=array[i-1];

grade[i]=2;

}

else if(array[i]>=70 && array[i]<80){

grade[i-1]=array[i-1];

grade[i]=3;

}

else{

grade[i-1]=array[i-1];

grade[i]=0;

}

}

for(i=0;i<size;i++)

printf("%d\n",grade[i]);

getchar();

getchar();

}