**91.Odd Even Average**

The Owner of a block visited the Layout and found that he has some plot numbers of his own and some are odd numbers and some are even numbers. He is maintaining the details in a file in the system. For the password protection our owner has followed one formula. He calculated the sum of his even numbers plot and sum of odd numbers plot and found the average of those two and he used that average as his password for the details file. Find the password that our owner has arrived.

Include a function named **avgOddEvenSum** that accepts 2 arguments and returns a float. 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 a float that corresponds to the average of 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+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 a floating point number that corresponds to the average. It is displayed correct to 2 decimal places.

Assume that the maximum size of the array is 20.

**Sample Input 1:**

5

1

2

3

4

5

**Sample Output 1:**

7.50

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

float avgOddEvenSum(int[],int);

int main(){

int n=0,i,flag=0,input[20];

float avg=0.0f;

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)

flag=1;

}

if(flag==1){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

avg = avgOddEvenSum(input,n);

printf("%0.2f",avg);

getchar();

getchar();

return 0;

}

float avgOddEvenSum(int array[],int size){

float avg;

int i,sumodd=0,sumeven=0;

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

if(array[i]%2==0)

sumeven=sumeven+array[i];

else

sumodd=sumodd+array[i];

}

avg = float(sumeven+sumodd)/2.0f;

return avg;

}

**92.convertToBinary**

Read the question carefully and follow the input and output format.  
  
Kate is a military officer. He needs to send top-secret code to the other soldiers in a different place. He need to encrypt it. We need to write a function to convert the given decimal number to the corresponding binary number.  
  
Input consist of single integer. Output consists of binary number.  
  
1) Print "Number too small" when input is negative.  
2) Print "Number too large" when input value is greater than 100.  
  
Include a function named convertToBinary(int num) whose return type is void.  
  
**Sample Input 1:**  
12  
  
**Sample Output 1:**  
1100  
  
**Sample Input 2:**  
101  
  
**Sample Output 2:**  
Number too large

#include<stdio.h>

#include<stdlib.h>

void convertToBinary(int num);

int main(){

int number=0;

scanf("%d",&number);

if(number<0)

printf("Number too small");

else if(number>100)

printf("Number too large");

else

convertToBinary(number);

return 0;

}

void convertToBinary(int num){

int a[20],i=0;

while(num>0)

{

a[i]=num%2;

i++;

num=num/2;

}

for(int j=i-1;j>=0;j--)

printf("%d",a[j]);

getchar();

getchar();

}

**93.reverseNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to find  the reverse of a given input integer  
  
**Input and Output Format :**  
Input consists of an integer, n.  Output consist of the reverse of the number n.  
  
Print "Number too large" when the given input number is greater than 32767  
Print "Number too small" when the given input numbers is a negative number.  
  
Include a function named reverseNumber(int num) whose return type is integer, the reverse of n.  
  
**Sample Input 1:**  
1234  
  
**Sample Output 1:**  
4321  
  
**Sample Input 2:**  
326357  
  
**Sample Output 2:**  
Number too large

#include<stdio.h>

#include<stdlib.h>

int reverseNumber(int num);

int main(){

int number=0,revnumber=0;

scanf("%d",&number);

if(number<0)

printf("Number too small");

else if(number>32767)

printf("Number too large");

else{

revnumber = reverseNumber(number);

printf("%d",revnumber);

}

getchar();

getchar();

return 0;

}

int reverseNumber(int num){

int reverse=0;

while (num != 0)

{

reverse = reverse \* 10;

reverse = reverse + num%10;

num=num/10;

}

return reverse;

}

**94.Sum of Prime Cubes**

Given an input integer n, write a program to find the sum of the cubes of the prime numbers upto n. (including n)

Please note that 1 is neither prime nor composite.

Include a function named **sumCubeOfPrime** that accepts an integer argument and returns an integer. The function returns -1 if the input is a negative number or if it is greater than 3000.

If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer.

Output consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

5

**Sample Output 1:**

160

**Sample Input 2:**

-241

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

50000

**Sample Output 3:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int sumCubeOfPrime(int);

int main(){

int number,result=0;

scanf("%d",&number);

result = sumCubeOfPrime(number);

if(result==-1)

printf("Invalid input");

else

printf("%d",result);

getchar();

getchar();

return 0;

}

int sumCubeOfPrime(int n){

int i,count=0,num,sum=0;

if(n<0 || n>3000)

sum=-1;

else{

for(num = 1;num<=n;num++){

count=0;

for(i=1;i<=num;i++){

if(num%i==0)

count++;

}

if(count==2)

sum=sum+(num\*num\*num);

}

}

return sum;

}

**95.Negative Elements**

Write a program to remove all the negative elements of the input array , sort the positive elements and stores them in an output array .

Include a function named **eliminateNegatives** 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 and the size of the output array is stored in a global variable named output2.

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.

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:**

8

1

6

3

5

-6

8

-15

9

**Sample Output 1:**

1

3

5  
  
6

8

9

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

 #include<stdio.h>

#include<stdlib.h>

int output1[20];

void eliminateNegatives(int[],int);

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]);

eliminateNegatives(input1,size);

return 0;

}

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

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

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

if(array[i]>0){

output1[k]=array[i];

k++;

}

}

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

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

if(output1[i]>output1[j]){

temp = output1[i];

output1[i]=output1[j];

output1[j]=temp;

}

}

}

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

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

getchar();

getchar();

}

**96.Savings Calculation**

Jim got salary for this month and he spends 50% of his salary for food and 20% of his salary for travel. If the number of days he worked is 31 he gets a bonus of Rs.500. Write a program to find how much he can save in his pocket after spending all these?

Include a function named **calculateSavings** that accepts 2 integer arguments and returns a float. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days Jim has worked. The function returns a float that corresponds to the amount that Jim could save.

Print Invalid Input and terminate the program in the following cases:

1. Basic salary is greater than 9000
2. Number of working days is greater than 31
3. Basic salary is negative
4. Number of working days is negative

**Input and Output Format:**

Input consists of 2 integers. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days he has worked.

Output consists of a single float that corresponds to Jim's savings. Jim's savings is displayed correct to 2 decimal places.

**Sample Input 1:**

7000

30

**Sample Output 1:**

2100 .00

**Sample Input 2:**

50000

**Sample Output 2:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

float calculateSavings(int,int);

int main(){

int basic=0,days=0;

float save=0.0f;

scanf("%d%d",&basic,&days);

if(basic>9000 || days>31 || basic<0 || days<0){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

save = calculateSavings(basic,days);

printf("%0.2f",save);

getchar();

getchar();

return 0;

}

float calculateSavings(int basic,int days){

float spent=0.0f,savings=0.0f;

spent=(float)(basic\*70)/100;

if(days==31)

savings = basic+500-spent;

else

savings = basic-spent;

return savings;

}

**97.studentMarks**

Read the question carefully and follow the input and output format.  
  
The Given input is of the format XXXYY , where XXX is Id , YY is marks. Write a code to display the id and marks separately as given in the output formats. [Refer sample input and output]  
  
**Input and Output Format :**  
Input consists of a number. Refer sample output for output format.  
  
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 studentMarks(int number) whose return type is void.  
  
**Sample Input 1:**  
12345  
  
**Sample Output 1:**  
123  
45  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

void studentMarks(int number);

int main(){

int number=0,result;

scanf("%d",&number);

if(number<0)

printf("number too small");

else if(number>32767)

printf("number too large");

else

studentMarks(number);

getchar();

getchar();

return 0;

}

void studentMarks(int number){

int id=0,mark=0,rem=0,i=1,flag=1,j=1;

while(number!=0){

if(flag<=2){

rem=number%10;

mark=mark+(rem\*i);

i=i\*10;

number=number/10;

flag++;

}

else{

rem=number%10;

id=id+(rem\*j);

j=j\*10;

number=number/10;

flag++;

}

}

printf("%d\n%d",id,mark);

getchar();

getchar();

}

**98.maximumSum**

Read the question carefully and follow the input and output format.  
  
Given an Integer array, find out sum of Even and odd Numbers individually and find the maximum.  
  
**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 maximum of odd and even 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 maximumSum(int numbers[], int size) whose return type is an integer,.  
  
**Sample Input 1:**  
5  
12  
13  
14  
15  
16  
  
**Sample Output 1:**  
42  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid array size

#include<stdio.h>

#include<stdlib.h>

int maximumSum(int[],int);

int main(){

int n=0,i,flag=0,input[20],max=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 input");

getchar();

getchar();

exit(0);

}

max = maximumSum(input,n);

printf("%d",max);

getchar();

getchar();

return 0;

}

int maximumSum(int numbers[], int size){

int i=0,max=0,evensum=0,oddsum=0;

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

if(numbers[i]%2==0)

evensum = evensum + numbers[i];

else

oddsum = oddsum + numbers[i];

if(evensum>oddsum)

max = evensum;

else

max=oddsum;

}

return max;

}

**99.newNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to find the difference between consecutive digits in the given input integer and display it.  
  
**Input and Output Format:**  
Input consists of an integer and output the difference between the consecutive digits.  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
  
Include a function named newNumber(int number) that returns a integer  
  
**Sample Input 1:**  
1325  
**Sample Output 1:**  
213  
  
**Sample Input 2:**  
-13  
**Sample Output 2:**  
Number too small

#include<stdio.h>

#include<stdlib.h>

int newNumber(int number);

int main(){

int num=0,new\_num=0;

scanf("%d",&num);

if(num>32767)

printf("Number too large");

else if(num<0)

printf("Number too small");

else{

new\_num=newNumber(num);

printf("%d",new\_num);

}

getchar();

getchar();

return 0;

}

int newNumber(int number){

int rem,i=1,rem1=0,diff=0,digit=0;

while(number>10){

rem=number%10;

number=number/10;

rem1=number%10;

if(rem>rem1)

diff=rem-rem1;

else

diff=rem1-rem;

digit=digit+(diff\*i);

i=i\*10;

}

return digit;

}

**100.nonWorkingDoctors**

Read the question carefully and follow the input and output format.  
  
A doctor survey results information is stored in 2 arrays. First array represents all doctors ids (working and non -working both). Second array represents only working doctor's id . Please find the doctor ids who are not working .  
  
**Input and Output Format :**  
First line of input corresponds to n1, the size of first array and next n1 lines correspond to the elements of the first array. The next line corresponds to n2, the size of second array and next n2 lines correspond to the elements of the second array  
Output is the id's of doctor who are not working  
  
Print "Invalid array size" when size of the array is a negative number and terminate the program  
Print "Invalid id" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named nonWorkingDoctors(int total[],int working[],int n,int m) whose return type is void.  
  
**Sample Input 1:**  
7  
7  
2  
3  
4  
5  
6  
1  
3  
3  
4  
5  
**Sample Output 1:**  
7  
2  
6  
1  
  
  
**Sample Input 2:**  
7  
7  
2  
3  
4  
5  
6  
-1  
**Sample Output 2:**  
Invalid id

#include<stdio.h>

#include<stdlib.h>

int output1[20];

void nonWorkingDoctors(int total[],int working[],int n,int m);

int main(){

int size1=0,size2=0,input1[20],input2[20],i=0;

scanf("%d",&size1);

if(size1<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<size1;i++){

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

if(input1[i]<0){

printf("Invalid id");

getchar();

getchar();

exit(0);

}

}

scanf("%d",&size2);

if(size2<0){

printf("Invalid array size");

getchar();

getchar();

exit(0);

}

for(i=0;i<size2;i++){

scanf("%d",&input2[i]);

if(input2[i]<0){

printf("Invalid id");

getchar();

getchar();

exit(0);

}

}

nonWorkingDoctors(input1,input2,size1,size2);

return 0;

}

void nonWorkingDoctors(int total[],int working[],int n,int m){

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

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

flag=0;

for(j=0;j<m;j++){

if(total[i]==working[j])

flag=1;

}

if(flag==0)

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

}

getchar();

getchar();

}