**21.Sum of Prime Numbers**

Write a program to find the sum of the prime numbers present in the given input array.

Include a function named**sumPrime**that accepts 2 arguments and returns an int. The first argument is a pointer to 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 prime numbers in the input 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.

Please note that 1 is neither prime nor composite.

**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.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

5

2

4

8

9

11

**Sample Output 1:**

13

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int sumPrime(int\*,int);

int output1[20];

int main(){

int n=0,flag=0,i,input[20],sum=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)

flag=1;

}

if(flag==1){

printf("Invalid Input");

getchar();

getchar();

exit(0);

}

sum = sumPrime(input,n);

printf("%d",sum);

getchar();

getchar();

return 0;

}

int sumPrime(int \*a,int n){

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

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

count=0;

for(j=1;j<=\*a;j++){

if((\*a)%j==0)

count++;

}

if(count==2)

sum=sum+(\*a);

}

return sum;

}

**22.Leap Year**

Write a program to find whether the given input year is a Leap Year.

Include a function named **checkLeapYear** that accepts an integer and returns an integer. The function returns

1. 1 if the input is a Leap Year
2. 0 if the input is not a Leap Year
3. -1 if the input is a negative number

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

2000

**Sample Output 1:**

yes

**Sample Input 2:**

1610

**Sample Output 2:**

no

**Sample Input 3:**

**-**2345

**Sample Output 3:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int checkLeapYear(int);

int main(){

int year=0,status=0;

scanf("%d",&year);

status = checkLeapYear(year);

if(status==-1){

printf("Invalid input");

}

else if(status==1){

printf("yes");

}

else{

printf("no");

}

getchar();

getchar();

return 0;

}

int checkLeapYear(int year){

int result;

if(year<0)

result=-1;

else if(year%4==0)

result=1;

else

result=0;

return result;

}

**23.secondLargest**

Read the question carefully and follow the input and output format.  
  
Write a function to find second largest number in the given input integer array.  
  
Assume that all elements in the input array are unique.  
  
**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 second largest.  
  
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 secondLargest(int array[], int size) whose return type is an integer, the second largest.  
  
**Sample Input 1:**  
5  
3  
342  
53  
2  
12  
  
**Sample Output 1:**  
53  
  
**Sample Input 2:**  
5  
3  
342  
53  
-2  
  
  
**Sample Output 2:**  
Invalid input

#include<stdio.h>

#include<stdlib.h>

int cricketer[20];

int secondLargest(int array[], int size);

int main(){

int n=0,largest=0,input[30],i,flag=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);

}

largest = secondLargest(input,n);

printf("%d",largest);

getchar();

getchar();

return 0;

}

int secondLargest(int array[], int size){

int largest=0,i,j,b;

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

{

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

{

if(array[i] < array[j])

{

b = array[j] ;

array[j] = array[i] ;

array[i] = b ;

}

}

}

return array[1];

}

**24.findFirstLargest**

Read the question carefully and follow the input and output format.  
  
Write a function to find the product of the first and the third largest numbers in the given input integer array.  
  
Input and Output Format :  
First line of input corresponds to n, the size of array and next n lines correspond to the elements of the array  
  
Print "Invalid array size" when size of the array is a negative number and terminate the program  
Print "Invalid number" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named findFirstLargest(int n, int array[]) that returns an integer, product of first and third largest numbers.  
  
**Sample Input 1:**  
5  
11  
241  
83  
8  
2  
  
**Sample Output 1:**  
2651  
  
**Sample Input 2:**  
3  
83  
-8  
**Sample Output 2:**  
Invalid number

#include<stdio.h>

#include<stdlib.h>

int cricketer[20];

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

int main(){

int n=0,largest=0,input[30],i,flag=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 Number");

getchar();

getchar();

exit(0);

}

largest = findFirstLargest(n,input);

printf("%d",largest);

getchar();

getchar();

return 0;

}

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

int largest=0,i,j,b,k;

/\*To remove duplicates\*/

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++;

}

}

/\*to sort array in desc order\*/

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

{

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

{

if(array[i] < array[j])

{

b = array[j] ;

array[j] = array[i] ;

array[i] = b ;

}

}

}

return (array[0] \* array[2]);

}

**25.generateCode**

Read the question carefully and follow the input and output format.  
  
In a game show everybody got one coupon with some code. They need to generate a code with only even numbers in that coupon. Find the answer.  
  
**Input and Output Format :**  
Input consists of  an integer. Output consist of an integer, which is the generated code.  
  
1) Print "Number too small" when the given input number is a negative number.  
2) Print "Number too large" when the given input number is greater than 32767.  
3) Print 0 If the coupon does not contain any even numbers.  
  
Include a function named generateCode(int coupon) whose return type is an integer, which is the generated code.  
  
**Sample Input 1:**  
4352  
  
**Sample Output 1:**  
42  
  
**Sample Input 2:**  
1357  
  
**Sample Output 2:**  
0  
  
**Sample Input 3:**  
-1357  
  
**Sample Output 3:**  
Number too small

#include<stdio.h>

#include<stdlib.h>

int generateCode(int coupon);

int main(){

int coupon=0,code=0;

scanf("%d",&coupon);

if(coupon<0)

printf("Number too small");

else if(coupon>32767)

printf("Number too large");

else{

code = generateCode(coupon);

if(code==0)

printf("%d",0);

else

printf("%d",code);

}

getchar();

getchar();

return 0;

}

int generateCode(int coupon){

int res=0,rem=0,i=1;

while(coupon!=0){

rem=coupon%10;

if(rem%2==0){

res=res+(rem\*i);

i=i\*10;

}

coupon = coupon/10;

}

return res;

}

**26.calculateBonus**

Read the question carefully and follow the input and output format.  
  
Given the basic salary as input, write a program to calculate the bonus and display it.  
  
The bonus will be calculated based on the below category.  
Basic>20000 bonus=17%of basic+1500  
Basic>15000 bonus=15%of basic+1200  
Basic>10000 bonus=12%of basic+1000  
for rest =8%of basic+500  
  
**Input and Output Format :**  
  
First line of input consists of n, the basic salary.  
Output is a single integer that displays the bonus.  
  
Print "Number too large" when the given input numbers is greater than 32767 .  
Print "Number too small" when the given input is a negative number.  
  
Include a function named calculateBonus(int basic) whose return type is an integer, the bonus.  
  
**Sample Input 1:**  
21000  
  
**Sample Output 1:**  
5070  
  
**Sample Input 2:**  
327678  
**Sample Output 2:**  
Number too large

#include<stdio.h>

#include<stdlib.h>

int calculateBonus(int basic);

int main(){

int salary=0,bonus=0;

scanf("%d",&salary);

if(salary<0)

printf("Number too small");

else if(salary>32767)

printf("Number too large");

else{

bonus = calculateBonus(salary);

printf("%d",bonus);

}

getchar();

getchar();

return 0;

}

int calculateBonus(int basic){

int res=0;

if(basic>20000)

res=(basic\*17/100)+1500;

else if(basic>15000)

res=(basic\*15/100)+1200;

else if(basic>10000)

res=(basic\*12/100)+1000;

else

res=(basic\*8/100)+500;

return res;

}

**27.Minimum of 3**

Write a program to find the minimum of 3 numbers.

Include a function named **findSmallest** that accepts 3 integer arguments and returns an integer that corresponds to the minimum value.

**Input and Output Format:**

Input consists of 3 integers.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

4

12

3

**Sample Output 1:**

3

#include<stdio.h>

#include<stdlib.h>

int findSmallest(int,int,int);

int main(){

int a,b,c,min=0;

scanf("%d%d%d",&a,&b,&c);

min = findSmallest(a,b,c);

printf("%d",min);

getchar();

getchar();

return 0;

}

int findSmallest(int a,int b,int c){

int res=0;

if(a<b && a<c)

res=a;

else if(b<a && b<c)

res=b;

else

res=c;

return res;

}

**28.Sort and Delete**

Write a program to delete the given number in the input array and then to sort the array.

Include a function named **sortAndDelete** that accepts 3 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 and the third argument is the array element to be deleted. The number of elements in the modified array is stored in the 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.

Please note that the elements in the array may not be unique.

**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 to be deleted.

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

8

10

4

8

8

**Sample Output 1:**

1

3

4

5

6

10

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

void sortAndDelete(int[],int,int);

int output1[20];

int main(){

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

flag=1;

}

if(flag==1){

printf("Invalid input");

getchar();

getchar();

exit(0);

}

scanf("%d",&element);

sortAndDelete(input,n,element);

return 0;

}

void sortAndDelete(int array[],int size,int element){

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

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

if(array[i]!=element){

output1[k]=array[i];

k++;

}

}

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

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

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

temp = output1[j];

output1[j] = output1[i];

output1[i] = temp;

}

}

}

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

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

}

getchar();

getchar();

}

**29.generateNewNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to generate new number from the given input based on following conditions.  
  
(i) Even digit should be replaced by next Even digit.  
(ii) Odd digit should be replaced with next Odd digit  
  
**Input and Output Format :**  
Input consists of an integer. Output is also an integer.  
  
1) Print "Number too small" when any of given input numbers is a negative number.  
2) Print "Number too large" when any of given input numbers is greater than 32767.  
  
Include a function named generateNewNumber(int number) whose return type is an integer, which is the replaced number.  
  
**Sample Input 1:**  
123  
  
**Sample Output 1:**  
345  
  
**Sample Input 2:**  
32768  
  
**Sample Output 2:**  
Number too large

#include<stdio.h>

#include<stdlib.h>

int generateNewNumber(int number);

int main(){

int number=0,new\_number=0;

scanf("%d",&number);

if(number<0)

printf("Number too small");

else if(number>32767)

printf("Number too large");

else{

new\_number = generateNewNumber(number);

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

}

getchar();

getchar();

return 0;

}

int generateNewNumber(int number){

int res=0,rem=0,i=1;

while(number!=0){

rem=number%10;

if(rem%2==0){

res=(rem+2)\*i+res;

i=i\*10;

}

else

{

res=(rem+2)\*i+res;

i=i\*10;

}

number = number/10;

}

return res;

}

**30.findMileage**

Read the question carefully and follow the input and output format.  
  
Given the cubic capacity(CC) of a bike. Write a function to return the mileage/liter for the given Cubic Capacity(CC). The mileage will be calculated as follows:  
  
if CC is between 100 and 125, mileage is 75  
if CC is between 126 and 135, mileage is 70  
if CC is between 136 and 150, mileage is 60  
if CC is between 151 and 200, mileage is 50  
if CC is between 201 and 220, mileage is 35  
  
First line of input consists of an integer that corresponds to CC of a bike. Output consist of an integer, which is the mileage.  
  
Print "Number too large" when the given input CC is greater than 220.  
Print "Number too small" when the given input CC is less than 100.  
  
Include a function named findMileage(int cc) whose return type is an integer, which is the mileage.  
  
**Sample Input 1:**  
1  
  
**Sample Output 1:**  
Number too small  
  
**Sample Input 2:**  
160  
  
**Sample Output 2:**  
50

#include<stdio.h>

#include<stdlib.h>

int findMileage(int cc);

int main(){

int cc=0,mileage=0;

scanf("%d",&cc);

if(cc<100)

printf("Number too small");

else if(cc>220)

printf("Number too large");

else{

mileage = findMileage(cc);

printf("%d",mileage);

}

getchar();

getchar();

return 0;

}

int findMileage(int cc){

int mil=0;

if(cc>=100 && cc<=125)

mil=75;

else if(cc>=126 && cc<=135)

mil=70;

else if(cc>=136 && cc<=150)

mil=60;

else if(cc>=151 && cc<=200)

mil=50;

else if(cc>=201 && cc<=200)

mil=35;

else

;

return mil;

}