Q1 Write a program in C to generate the electricity bills for 5 customers and find out the total amount paid by the all customers as per following billing plan to be paid by the customer (as per consumed units):

If consumed units are 0-200 then electricity charges are NIL.

If consumed units are 0-300 then electricity charges are Rs. 3 per unit.

If consumed units are 301-500 then electricity charges are Rs. 3 per unit for 1 to 300 units and Rs. 5 per unit for 301 to 500 units.

If consumed units are more than 500 then electricity charges are Rs. 3 per unit for 1 to 300 units, Rs. 5 per unit for 301 to 500 units and Rs. 7 per unit for 501 and above.

**Input:**

First five lines of the input contains two space separated values n and m, representing the previous meter reading and the current meter readings respectively of each customer.

**Output:**

Output shows one value reflecting the total electricity bill of all the customers

**Constraints:**

n<=m

Sample test Case:

|  |  |  |
| --- | --- | --- |
| S.No | Input | Output |
| STC1 | 234 696  213 897  345 756  23 876  343 1909 | Total amount paid: 20086 |

**Solution 16:**

#include<stdio.h>

int i;

int main()

{

int cu, pu, u, amt=0;

for(i=1;i<=5;i++)

{

// printf("Enter the previous and current consumed units for %d customer:", i);

scanf("%d %d", &pu, &cu);

u=cu-pu;

if(u<=200)

{

amt=amt+0;

}

else if(u<=300)

{

amt=amt+u\*3;

}

else if(u<=500)

{

int t=u-300;

amt=amt+300\*3+t\*5;

}

else

{

int t=u-500;

amt=amt+300\*3+200\*5+t\*7;

}

}

printf("%d", amt);

return(0);

}

**Hidden Test Cases:**

|  |  |  |
| --- | --- | --- |
| S.No. | Input | Output |
| TC1 | 200 340  400 900  200 1000  220 420  30 450 | 7400 |
| TC2 | 200 650  230 476  230 678  123 678  23 456 | 7878 |
| TC3 | 200 300  200 350  100 200  200 500  150 300 | 900 |
| TC4 | 250 400  500 600  400 800  150 300  279 400 | 1400 |
| TC5 | 2786 2345  234 675  234 786  21 345  345 789 | 6509 |

1. 2 Chitkara University organize a Football champions league in summer. All Player are selected in each group exept goalkeeper. Football management wants to select goalkeeper in each groups so they decided that goalkeeper is selected based on height in each groups. The criteria for becoming goalkeeper is to have maximum heights amongs all students in that group.

You have given a number n which denote number of students , and an array of size n (heights[i] ) which denote heights

Of students. Find the goalkeeper height

Note: All students must have different hights.

Input Format: Take an integer array of size (n)

Output Formate : Print maximum heights to find goalkeeper

Sample input 1:

4

2 1 5 3

Sample output 1:

5

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 8  1 2 3 4 5 6 7 8 | 8 |
| STC2 | 5  9 8 7 6 5 | 9 |
| STC3 | 1  5 | 5 |
| STC 4 | 7  2 1 5 3 9 4 6 | 9 |
| STC 5 | 4  2 6 12 5 | 12 |

**Sample Test Cases:**

**Solution 17:**

#include<stdio.h>

int i;

int main(){

int n; scanf("%d",&n);

int arr[n];

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

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

}

int max=0;

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

if(max<arr[i])

max=arr[i];

}

printf("%d",max);

}

**Q3. Alternate Digit Sum**

John is a creative genius(or so he likes to call himself). He is given a number and add its digit with alternate signs (i. e.

+ and -) . For eg. if the given digit is 896 john will do +8 + (-9) + (+6) and return 5 as answer. First sign will be positive.

Write a C program to help Sahil achieve the same functionality in C.

Input Format:

takes a positive integer n as input

Output Format:

return the sum of its digits from left with alternate signs.

Sample input 1 : 8796

Sample output 1: 4

Explanation +8 + (-7) + 9 + (-6) = 4

Sample input 2 : 9999

Sample output 2: 0

Sample input 3: 10

Sample output 3: 1

**Sample Test Case:**

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 102 | 3 |
| STC2 | 19 | -8 |
| STC3 | 1000 | 1 |

**Solution 3:**

#include<stdio.h>

int main()

{

int n; scanf("%d",&n);

int dig=0;

int k=n;

while(k>0){

dig++;

k=k/10;

}

int sum=0,i;

if(dig%2==0)

i=1;

else

i=0;

while(n>0){

if(i%2==0)

sum=sum+n%10;

else

sum=sum-n%10;

n=n/10;

i++;

}

printf("%d", sum);

}

Q4.The amount of pocket money received by three siblings is stored at three different location inside main memory. Write a program to swap the values of the money received using call by reference.

**Input:**

500600700

**Output:**

700 500 600

Input contains three space separated values as the amount of pocket money received by three siblings.

output shows three space separated values as the swapped content against input supplied.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 100400900 | 900 100 400 |
| STC2 | 650150450 | 450 650 150 |
| STC3 | 5 7 8 | 8 5 7 |

**Solution 4:**

#include <stdio.h>

void swapNumbers(int \*x,int \*y,int \*z);

int main()

{

int e1,e2,e3;

scanf("%d",&e1);

scanf("%d",&e2);

scanf("%d",&e3);

swapNumbers(&e1,&e2,&e3);

printf("%d %d %d",e1,e2,e3);

return 0;

}

void swapNumbers(int \*x,int \*y,int \*z)

{

int tmp;

tmp=\*y;

\*y=\*x;

\*x=\*z;

\*z=tmp;

}

Q 5 A system is designed in such a way that it will store the roll number of students in even/odd fashion and then display the even/ odd pattern to the user individually. Initially enter all the roll numbers as a normal array. For this purpose, write a C program by passing the roll numbers using arrays to a function for printing even and odd lists separately.

**Input:**

6

101103104105107190

**Output:**

104, 190

101, 103, 105, 107

First line of the input reflects the total "n" number of rollnumbers that user have to enter. Second line contains n space separated values reflecting rollnumbers of individual students and is the mixture of even and odd vlaues.

The first line of output display the space separated rollnumbers reflecting even values only.

The second line of the output display the space separated rollnumbers reflecting odd values only as per the input supplied.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 4  1001100310421005 | 1042  1001 1003 1005 |
| STC2 | 7  1010103010401050105110701900 | 1010 1030 1040 1050 1070 1900  1051 |
| STC3 | 4  101 103 144 156 | 1044 156  101 103 |

**Solution 5:**

#include<stdio.h>

void checkEven(int e);

void checkOdd(int o);

static int n1=0, n2=0;

int main()

{

int arr[100], i, size;

scanf("%d", &size);

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

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

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

checkEven(arr[i]);

printf("\n");

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

checkOdd(arr[i]);

return 0;

}

void checkEven(int e)

{

if(e%2==0)

{

printf("%d ",e);

;

}

}

void checkOdd(int o)

{

if(o%2 != 0)

{

printf("%d ",o);

}

}

Q 6 **Abhi likes mathematics, he loves to solve problems. Recently he came across the concept of perfect square. A perfect square is an integer that can be expressed as square of another integer (smaller than or equal to the given integer).He was trying to find out how many perfect squares exist between two given integers, but got confused, help him out to find the solution.**

**Your task is to check all the numbers between a and b, and print only those numbers which are perfect square numbers.**

**Write a program that takes two numbers a and b as input and prints all the perfect square numbers that exist between a and b (both inclusive) in a new line.**

**Sample Input**:

120

**Sample Output**:

14916

Input shows two space separated integers representing starting point of the range and ending points respectively in which we have to display all perfect squares.

The output represents n space separated integers, where n is the total number of perfect squares existing in the input range.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 | Test Case 4 | Test Case 5 |
| Input | 10100 | 10 40 | 40 80 | 1050 | 90 150 |
| Output | 162536496481100 | 162536 | 4964 | 16253649 | 100121144 |

**Solution:**

#include <stdio.h>

intisPerfectSq(intnum)

{

for(int i=0; i<=num; i++)

{

if(i\*i == num)

{

return 1;

}

}

return 0;

}

int main()

{

inta,b;

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

for(int i=a; i<=b; i++)

{

if(isPerfectSq(i)==1)

{

printf("%d ",i);

}

}

return 0;

}

Q 7. Statement: Virat received the result of "n" students. His task is to find out the highest marks in the class along with the number of students who scored the highest marks. For example, if a class has n=5 students and the marks of the students are 22, 33, 23, 33, and 33 respectively then the output of the work performed by Virat must be 33 and 3. highest marks are 33 and 3 students scored 33 marks.

Input format

5

22 33 23 33 33

Output format

33 3

In the above example, the First line of the input is representing the total number of students i.e 5. The second line represents the marks obtained by these five students by five space-separated values. The output display two space-separated values, the first value of output represents the highest marks obtained by the students and the second value of output represents the number of students who obtained the highest marks.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Output** |
| **TC1** | **5**  **22 33 23 33 33** | **33 3** |
| **TC2** | **6**  **55 66 44 33 12 22** | **66 1** |
| **TC3** | **3**  **77 77 77** | **77 3** |
| **TC4** | **6**  **55 76 44 76 12 22** | **76 2** |
| **TC5** | **7**  **55 66 44 33 12 22 78** | **78 1** |

**Solution:**

**#include<stdio.h>**

**int main()**

**{**

**int a[100],n,i;**

**scanf("%d",&n);**

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

**{**

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

**}**

**int big=0;**

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

**{**

**if(a[i]>big)**

**{**

**big=a[i];**

**}**

**}**

**int count=0;**

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

**{**

**if(a[i]==big)**

**{**

**count=count+1;**

**}**

**}**

**printf("%d %d",big,count);**

**return 0;**

**}**

**Q 8**. A faculty needs to find out the highest and the lowest marks from the marks obtained by "n" students. Write a program that will take input as two lines. The first line of input shows the total number of students in the class as "n". The second line will consider "n" space-separated values as the marks obtained by each student. The program will display output as two space-separated values representing the highest and lowest marks respectively.

Input:

4

22 33 11 44

Output:

44 11

The first line of input shows the total number of students as 4. The second line of input shows 4 values as 22,33,11,44. The output shows two space-separated values 44 as the highest among the different inputs entered and 11 as the lowest among the input values.

Test Cases:

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Output** |
| **Test\_1** | **4**  **22 33 11 55** | **55 11** |
| **Test\_2** | **5**  **10 12 13 14 22** | **22 10** |
| **Test\_3** | **6**  **11 22 44 88 99 66** | **99 11** |
| **Test\_4** | **3**  **1 44 5** | **44 1** |
| **Test\_5** | **7**  **21 22 44 88 98 66** | **98 21** |

**solution:**

**#include<stdio.h>**

**int main()**

**{**

**int n,i, a[100];**

**int big=0, small;**

**scanf("%d",&n);**

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

**{**

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

**}**

**small=a[0];**

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

**{**

**if(big<a[i])**

**{**

**big=a[i];**

**}**

**if(small>a[i])**

**{**

**small=a[i];**

**}**

**}**

**printf("%d %d",big,small);**

**return 0;**

**}**

Q 9.Rina is playing a game in which she is throwing a dice any no. of times. A dice contain a number from 1 to 6. After that, she is counting which number is occurring and how many times.There should be no space before or after a colon(:) in the output.

**Input:**

Throw dice any number of times.

The number occurs on dice.

**Constraints:**

1<=**n**<=6

**Output:**

Count how many times a number is occurring.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 5  1 6 3 6 1 | 1:2  6:2  3:1 |
| STC2 | 5  4 5 6 5 4 | 4:2  5:2  6:1 |

**Solution:**

#include <stdio.h>

int main()

{

int a[100], freq[100];

int size, i, j, c;

scanf("%d", &size);

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

{

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

freq[i] = -1;

}

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

{

c = 1;

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

{

if(a[i]==a[j])

{

c++;

freq[j] = 0;

}

}

if(freq[i] != 0)

{

freq[i] = c;

}

}

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

{

if(freq[i] != 0)

{

printf("%d:%d\n", a[i], freq[i]);

}

}

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 6  1 2 1 3 5 3 | 1:2  2:1  3:2  5:1 |
| TC2 | 5  5 3 5 5 5 | 5:4  3:1 |
| TC3 | 7  2 3 2 3 2 1 1 | 2:3  3:2  1:2 |
| TC4 | 6  6 5 6 5 2 5 | 6:2  5:3  2:1 |
| TC5 | 5  5 5 5 5 | 5:5 |

**Q 10. Write a C program to print the Fibonacci numbers using functions based on the user input for the length of the printed numbers.**

**Input:**

6

**Output:**

0 1 1 2 3 5

Input contains a single integer reflecting how many elements of Fibonacci series that the user wants to print. In the above example input is given as 6, means first six elements of the Fibonacci series we have to print. The output is displaying the first six elements of Fibonacci series by showing first six space separated values of Fibonacci series.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 20 | 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 |
| STC2 | 9 | 0 1 1 2 3 5 8 13 21 |

**Solution:**

#include <stdio.h>

// fibonacci() funtion definition

void fibonacci(int num)

{

if (num == 1)

{

printf("0 ");

return; // retuning 0, if condition meets

}

else if (num == 2)

{

printf("0 1 ");

return ;

}

else

{

int a=0;

int b=1;

int feb;

printf("%d %d ",a,b);

for (int i=1;i<=num-2;i++)

{

feb=a+b;

printf("%d ",feb);

a=b;

b=feb;

}

return;

}

}

int main()

{

int num;

scanf("%d", &num);

fibonacci(num);

return 0;

}

**Q11 Mr. Sharma is a bank manager. Bank sets an alarm when someone will put in a reverse ATM pin so that an alarm can be sent to the police. A few customers already complained that this is not working.**

**So, Mr. Sharma asked for help from their IT personnel to write a C program where an input 4-digit pin will be taken in an array and then it will be reversed and finally the reversed output should be displayed.**

**Input:**

1 2 3 4

**Output:**

4 3 2 1

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 1 2 3 4 | 4 3 2 1 |
| STC2 | 6 7 8 9 | 9 8 7 6 |

**Solution 17:**

#include <stdio.h>

#include <stdlib.h>

int main(){

int n = 4;

int arr[4],temp;

for (int j=0;j<n;j++)

{

scanf("%d",&arr[j]);

}

for(int i = 0; i<n/2; i++)

{

temp = arr[i];

arr[i] = arr[n-i-1];

arr[n-i-1] = temp;

}

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

{

printf("%d ", arr[i]);

}

return 0;

}

**Q 12 Rohan is playing a game. The game has the following rules. He has to pick a number N( N can be any positive integer) and has to find all the Armstrong numbers that come between 1 to N.**

if an Armstrong is a number that is equal to the sum of the cube of its digits.

Rohan is facing some difficulty, help him to find the solution.

Write a C program using a function that takes a number N as an argument and prints all the Armstrong numbers between 1 to N (both inclusive).

**Input: 10**

**Output:**

0

1

The above output will be displayed when the user will input N=10.

TEST CASE 1:

410

**Out put of TEST CASE 1:**

0 1 153 370 371 407

**TEST CASE 2:**

10

**Output of TEST CASE 2:**

0 1

**SOLUTION:**

#include<stdio.h>

void printAmstong(int N)

{

int num, cubeSum, rem;

for (int i = 0; i<= N; i++)

{

cubeSum = 0;

num = i;

while (num != 0)

{

rem = num % 10;

num /= 10;

cubeSum += (rem \* rem \* rem);

}

if (i == cubeSum)

{

printf("%d ", i);

}

}

}

int main()

{

int N;

scanf("%d", &N);

printAmstong(N);

return 0;

}

Q 13 Karan Batra is an assistant professor in Chitkara University. He has to show the report of his students' performance to the Beta Cluster Dean. For data analysis he tried to make a C program which showed the division of students according to the marks secured by them. He made 5 groups. Each student must fall in one of the groups according to marks obtained by the student.

|  |  |
| --- | --- |
| **Group** | **Marks Obtained Between** |
| **1** | **0-39** |
| **2** | **40-49** |
| **3** | **50-59** |
| **4** | **60-79** |
| **5** | **80-100** |

Explanation: Input: Ram will first enter the total no. of students for which he wishes to enter the marks. And then he will enter the marks (0<=marks<=100)

Output: The 5 groups must be displayed on the screen which will reveal the no. of students falling in that specific group.

**Sample test Case 1**

3 //Total number of students

34

56

89

**Sample Output 1**

1. 1 //group number and count of students who are lying in group 1
2. 0 // group number and count of students who are lying in group 2
3. 1 // group number and count of students who are lying in group 3
4. 0 // group number and count of students who are lying in group 4
5. 1 // group number and count of students who are lying in group 5

**Sample test Case 2**

4 //Total numbers of students

55

57

76

67

**Sample Output 2**

1. 0 // count of students who are lying in group 1
2. 0 // count of students who are lying in group 2
3. 2 // count of students who are lying in group 3
4. 2 //count of students who are lying in group 4
5. 0 // count of students who are lying in group 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **TC1** | **TC2** | **TC3** | **TC4** | **TC5** |
| **Input** | 2  54  25 | 3  45  45  92 | 5  34  45  67  65  78 | 4  56  54  78  89 | 3  32  32  78 |
| **Output** | 1 1  2 0  3 1  4 0  5 0 | 1 0  2 2  3 0  4 0  5 1 | 1 1  2 1  3 0  4 3  5 0 | 1 0  2 0  3 2  4 1  5 1 | 1 2  2 0  3 0  4 1  5 0 |

**Solution**

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int arr[100],i,n,c1=0,c2=0,c3=0,c4=0,c5=0;

scanf("%d",&n);

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

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

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

{

if(arr[i]>=0 && arr[i]<=39)

c1++;

else if(arr[i]>=40 && arr[i]<=49)

c2++;

else if(arr[i]>=50 && arr[i]<=59)

c3++;

else if(arr[i]>=60 && arr[i]<=79)

c4++;

else if(arr[i]>=80 && arr[i]<=100)

c5++;

}

i=1;

printf("%d %d\n",i++,c1);

printf("%d %d\n",i++, c2);

printf("%d %d\n",i++, c3);

printf("%d %d\n",i++, c4);

printf("%d %d\n",i++, c5);

return 0;

}

Q 14. Rima is first year student of Computer Science and Engineering in University. She always got confused with the palindrome numbers. Help her to solve the problem by checking the entered number is palindrome number or not.

**Sample Input 1**

1221

**Sample Output 1**

1221 Palindrome //Display the reversed number first and then display a message either Palindrome or Not Palindrome

**Sample Input 2**

1212

**Sample Output 2**

2121 Not Palindrome

**Test Cases**

|  |  |  |
| --- | --- | --- |
|  | **Input Number** | **Output** |
| **TC1** | 1111 | 1111 Palindrome |
| **TC2** | 5252 | 2525 Not Palindrome |
| **TC3** | 8998 | 8998 Palindrome |
| **TC4** | 9182 | 2819 Not Palindrome |
| **TC5** | 7678 | 8767 Not Palindrome |

**Solution**

#include <stdio.h>

int main()

{

int n, rev = 0, rem, num;

scanf("%d", &n);

num = n;

while (n != 0)

{

rem = n % 10;

rev = rev \* 10 + rem;

n /= 10;

}

if (num == rev)

printf("%d Palindrome", rev,num);

else

printf("%d Not Palindrome",rev,num);

return 0;

}

**Q15** **Problem Statement: Given an array of integers, *marks*, denoting the marks scored by students in a class. The alternating elements *marks0*, *marks2*, *marks4*, and so on denote the marks of boys. Similarly, *marks1*, *marks3*, *marks5,* and so on denote the marks of girls. The array name *marks* store the base address of that array.**

**The program must return the sum of marks for boys if gender 1 or returns the sum of marks for girls if gender = 2.**

**Input Format**

* The first line contains ***number of students***, denoting the number of students in the class, hence the number of elements in ***marks***.
* Each of the ***number of students,*** subsequent lines contain ***marksi***.
* The next line contains ***gender***.

**Sample Input**:

3 // number of students

3 // marks of students

2

5

1 // gender

**Sample Output**:

8 //***marks*** = [3,2,5] and ***gender*** = 1

So, ***marks0 + marks2*** = 3+5 = 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 | Test Case 4 |
| Input | 5  5  7  8  9  10  1 | 5  1  2  3  4  5  2 | 1  4  2 | 1  9  1 |
| Output | 23 | 6 | 0 | 9 |

**Solution:**

#include <stdio.h>

int main() {

int num,i;

scanf("%d", &num);

int marks[10];

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

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

}

int gender;

scanf("%d", &gender);

int sum = 0;

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

{

if(gender==1)

{

if (i%2==0)

{

sum += marks[i];

}

}

else if (gender==2)

{

if (i%2==1)

{

sum += marks[i];

}

}

}

printf("%d",sum);

return 0;

}

**Q 16**  **Rohan is teaching artificial intelligence subject. He want to keep the record of marks scored by students in this subject in a sorted way. Help rohan to sort n numbers stored in an array.**

**Input Explanation:**

First line takes the number of elements in an array.

Second line onwards takes the elements of array.

**Output Explanation:**

First line shows the elements of array.

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| **STC1** | **3**  **23**  **10**  **15** | **The given array is:23 10 15**  **Sorted array is:10 15 23** |
| **TC1** | **3**  **12**  **8**  **5** | **The given array is:12 8 5**  **Sorted array is:5 8 12** |
| **TC2** | **3**  **15**  **18**  **5** | **The given array is:15 18 5**  **Sorted array is:5 15 18** |
| **TC3** | **3**  **12**  **8**  **5** | **The given array is:12 8 5**  **Sorted array is:5 8 12** |
| **TC4** | **2**  **12**  **8** | **The given array is:12 8**  **Sorted array is:8 12** |
| **TC5** | **2**  **14**  **8** | **The given array is:14 8**  **Sorted array is:8 14** |

**Solution**

#include <stdio.h>

int countSort(int arr1[], int n, int exp)

{

int output[n];

int i, ctr[n] ;

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

ctr[i] = 0;

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

ctr[ (arr1[i]/exp)%n ]++;

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

ctr[i] += ctr[i - 1];

for (i = n - 1; i >= 0; i--)

{

output[ctr[ (arr1[i]/exp)%n] - 1] = arr1[i];

ctr[(arr1[i]/exp)%n]--;

}

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

arr1[i] = output[i];

}

void sortArray(int arr1[], int n)

{

countSort(arr1, n, 1);

countSort(arr1, n, n);

}

void printBothArr(int arr1[], int n)

{

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

printf("%d ",arr1[i]);

}

int main()

{

int i, m;

scanf(“%d”, &m)

int arr1[m] ;

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

scanf("%d ",&arr1[i]);

int n = sizeof(arr1)/sizeof(arr1[0]);

printf("The given array is:");

printBothArr(arr1, n);

sortArray(arr1, n);

printf("\nSorted array is:");

printBothArr(arr1, n);

return 0;

}

**Q 17 Sanya is learning about vowel and consonants. Her teacher gave an assignment to count total number of vowel or consonant in a string. Write a program that takes a string as input and find the number of vowels and consonants in that.**

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| **STC1** | **Welcome to Chitkara University** | **The total number of vowel in the string is : 11**  **The total number of consonant in the string is : 16** |
| **STC2** | **Fundamental of c programming** | **The total number of vowel in the string is : 8**  **The total number of consonant in the string is : 17** |
| **STC2** | **Hello World** | **The total number of vowel in the string is : 3**  **The total number of consonant in the string is : 7** |
| **STC2** | **Fundamental of c programming** | **The total number of vowel in the string is : 8**  **The total number of consonant in the string is : 17** |

**Solution**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define str\_size 100 //Declare the maximum size of the string

void main()

{

char str[str\_size];

int i, len, vowel, cons;

fgets(str, sizeof str, stdin);

vowel = 0;

cons = 0;

len = strlen(str);

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

{

if(str[i] =='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u' || str[i]=='A' || str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U')

{

vowel++;

}

else if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))

{

cons++;

}

}

printf("The total number of vowel in the string is:%d\n", vowel);

printf("The total number of consonant in the string is:%d", cons);

}

**Q18. Write a program in C to print the number of Palindrome numbers in the given range.**

**Input:**

One input as integer used to provide the lower range

Second input as integer used to provide the upper range

**Output:**

**Single** line print of each palindroms number from the given range

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 100  150 | 101 111 121 131 141 |
| STC2 | 150  200 | 151 161 171 181 191 |
| STC1 | 200  250 | 202 212 222 232 242 |
| STC2 | 250  300 | 252 262 272 282 292 |
| STC5 | 100  150 | 101 111 121 131 141 |
| STC6 | 150  200 | 151 161 171 181 191 |

**Solution 18:**

#include<stdio.h>

int main()

{

int num, rem, reverse\_num, temp, start, end, count;

scanf("%d",&start);

scanf("%d",&end);

for(num=start;num<=end;num++)

{

temp=num;

reverse\_num=0;

while(temp)

{

rem=temp%10;

temp=temp/10;

reverse\_num=reverse\_num\*10+rem;

}

if(num==reverse\_num)

{

printf("%d ",num);

count++;

}

}

}

**Q19:** The games development company "FunGames" has developed a balloon shooter game. The balloons are arranged in a linear sequence and each balloon has a number associated with it. The numbers on the balloons are in the Fibonacci series. In the game, the player shoots 'k' balloons. The player's score is the sum of numbers on the 'k' balloons. Write a program to generate the player's score.

Sample Test case

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Sample Input** | **Sample Output** |
| **1** | **7** | **20 // Sum of Fibonacci numbers upto 7 terms i.e. 0+1+1+2+3+5+8=20** |
| **2** | **8** | **33 // Sum of Fibonacci numbers upto 20 terms i.e. 0+1+1+2+3+5+8+13=33** |

Test case:

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Sample Input** | **Sample Output** |
| **1** | **9** | **54** |
| **2** | **10** | **88** |
| **3** | **5** | **7** |
| **4** | **4** | **4** |
| **5** | **15** | **986** |

**Source Code: 19**

#include<stdio.h>

int fibo(int n)

{ int i;

int fibs[n];

fibs[0] = 0;

fibs[1] = 1;

int sum=0;

for(i=2;i<n;i++)

{

fibs[i] = fibs[i-1]+fibs[i-2];

}

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

{

sum += fibs[i];

}

return sum;

}

int main()

{

int numBalloons;

scanf("%d",&numBalloons);

if (numBalloons==0)

{

printf("0");

}

else

{

int result = fibo(numBalloons);

printf("%d",result);

}

}

**Q20.**  **Write a program to print the marks of 5 subjects, display the total marks and average of the result, further check whether the average number is a prime number or not, with return value and argument using functions.**

Sample Test Case

|  |  |  |
| --- | --- | --- |
|  | TC1 | TC2 |
| Input | 50 51 52 53 54 | 71 71 71 71 71 |
| Output | Total:260  Average:52  52 is not a prime number | Total Marks:355  Average:71  71 is a prime number |

Test Cases: // Eliminated the statement from input test case : Enter the Marks:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TC1 | TC2 | TC3 | TC4 | TC5 |
| Input | 62 64 68 60 70 | 15 16 17 18 19 | 92 82 78 60 79 | 50 51 52 53 54 | 60 61 62 63 64 |
| Output | Total:324  Average:64  64 is not a prime number | Total:85  Average:17  17 is a prime number | Total:391  Average:78  78 is not a prime number | Total:260  Average:52  52 is not a prime number | Total:310  Average:62  62 is not a prime number |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Solution 20:**

#include<stdio.h>

#include<math.h>

int marklist(int,int,int,int,int);

int is\_prime(int);

int main()

{

int hin,eng,mat,sci,soc,tot,avg,ans;

scanf("%d%d%d%d%d",&hin,&eng,&mat,&sci,&soc);

ans=marklist(hin,eng,mat,sci,soc);

printf("\nTotal:%d",ans);

avg=ans/5;

printf("\nAverage:%d\n",avg);

if (is\_prime(avg)) {

printf("%d is a prime number\n",avg);

}

else

{printf("%d is not a prime number\n",avg);

}

return 0;

}

int is\_prime(int avg) {

if (avg<=1) {

return 0;

}

int i;

for (i = 2;i<= sqrt(avg); i++) {

if (avg % i ==0) {

return 0;

}

}

return 1;

}

int marklist(int hin,int eng,int mat,int sci,int soc)

{

int tot;

tot=hin+eng+mat+sci+soc;

return tot;

}

:

**Q21** **Problem Statement: Write a program in C to print the elements of an array in reverse order.**

**Sample Input**:

5 \\ Number of elements to store in the array

4 3 2 8 6 \\ Elements of the unsorted array

**Sample Output**:

6

8

2

3

4 \\ The elements of array in reverse order (2 3 4 5 7)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 | Test Case 4 | Test Case 5 |
| Input | 4  65 70 75 80 | 9  45 50 55 60 65 70 75 80 85 | 6  25 63 87 92 | 3  43 45 47 | 7  45 14 02 40 61 22 84 |
| Output | 80  75  70  65 | 85  80  75  70  65  60  55  50  45 | 92  87  63  25 | 47  45  43 | 84  22  61  40  2  14  45 |

**Solution:**

#include <stdio.h>

int main()

{

int n, i, arr1[15];

int \*pt;

// printf("\n\n Pointer : Print the elements of an array in reverse order :\n");

// printf("----------------------------------------------------------------\n");

// printf(" Input the number of elements to store in the array (max 15) : ");

scanf("%d",&n);

pt = &arr1[0]; // pt stores the address of base array arr1

// printf(" Input %d number of elements in the array : \n",n);

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

{

// printf(" element - %d : ",i+1);

scanf("%d ",pt);//accept the address of the value

pt++;

}

pt = &arr1[n - 1];

// printf("\n The elements of array in reverse order are :");

for (i = n; i > 0; i--)

{

printf("%d\n",\*pt);

pt--;

}

//printf("\n\n");

return 0;

}

**Q22.** **“If you are being robbed as you withdraw cash from an ATM (or are forced to against your will), do not fight back. Simply input your PIN code in reverse – for example, dial 4321 instead of 1234. As you do this, the ATM will secretly alert the police and make the machine appear to be malfunctioning”.**

**In 1994 Joseph Zinger, an American lawyer, filed for a patent on such a method. Zinger offered a reversed PIN as an emergency call from the bank to the police during ATM transection. The patent in question was granted to Zinger, yet the deployment did not go smoothly.**

**However, a bank decided to apply the same and ask you to write a C program where as input 4-digit pin will be taken in an array and then it will be reversed and finally the reversed output should be displayed.**

**Input:**

4-digit array.

**Output:**

Reversed pin number.

**Sample test Cases**

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 1 2 3 4 | 4 3 2 1 |
| STC2 | 60 70 80 90 | 90 80 70 60 |
| STC3 | 101 220 370 459 | 459 370 220 101 |
| STC4 | 33 45 55 91 | 91 55 45 33 |

**Solution 16:**

#include<stdio.h>

int main()

{

int n = 4;

int arr[4];

int i,j;

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

{

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

}

for(j = n-1; j>=0; j--)

{

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

}

}

**Q 23 Write a C program to display the n terms of odd natural number and their sum.**

**Sample Input**

Input number of terms: 10

**Sample Output**

The odd numbers are: 1 3 5 7 9 11 13 15 17 19

The sum of odd number upto 10 terms: 100

**Sample Input**

Input number of terms: 5

**Sample Output**

The odd numbers are: 1 3 5 7 9

The sum of odd number upto 5 terms: 25

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 2 | The odd numbers are: 1 3  The sum of odd number upto 2 terms: 4 |
| STC2 | -5 | The odd numbers are: (blank)  The sum of odd number upto -5 terms: 0 |

**Solution 16:**

#include <stdio.h>

void main()

{

int i,n,sum=0;

printf("Input number of terms : ");

scanf("%d",&n);

printf("\nThe odd numbers are :");

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

printf("%d ",2\*i-1);

sum+=2\*i-1;

}

printf("\nThe Sum of odd Natural Number upto %d terms : %d \n",n,sum);

}

**Q24** **Problem Statement: John’s mother, had taken some random numbers from a database. She wants to check John’s computing ability so she asked him to find those numbers that must consists of ten digits, where digits must lie in the range from 1 to 9 only. Then she also asked him to check whether the sum of the digits of that number is a palindrome or not. Can you help him to solve his problem?**

**Note: Input from user should not contain any 0, as digit.**

**Sample Input**:

3 // Input number of test cases

2587496356 //Input First number consist of 10 digits not contain, 0 as digit.

14785236 // Input second number consist of 8 digits not contain, 0 as digit.

2587496352 // Input third number consist of 10 digits not contain, 0 as digit.

**Sample Output**:

55, Number is Palindrome //Output, sum of number is 55, which is palindrome

INVALID NUMBER //Output invalid number

51, Number is not a Palindrome //Output, sum of number is 51, which is palindrome

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 | Test Case 4 | Test Case 5 |
| Input | 1  2587496354 | 2  111115555  9999999997 | 3  1  3698521478  987546321458 | 3  12345  3333333333  111111222222 | 2  9225132414  98745632547896 |
| Output | 53, Number is not a Palindrome | INVALID NUMBER  88, Number is Palindrome | INVALID NUMBER  53, Number is not a Palindrome  INVALID NUMBER | INVALID NUMBER  30, Number is not a Palindrome  INVALID NUMBER | 33, Number is Palindrome  INVALID NUMBER |

Solution:

#include<stdio.h>

void check (long int a);

int main ()

{

int t, i;

long n;

scanf("%d", &t);

for (i=0;i<t;i=i+1)

{

scanf("%ld", &n);

check(n);

}

}

void check(long a)

{

int i, c=0, rem;

int sum=0, revnum=0 ;

long num, num1, num2;

num=a;

int flag=1;

while(num!=0){

c=c+1;

rem=num%10;

if(rem==0)

{

flag=0;

break;

}

sum=sum+rem;

num=num/10;

}

if(c!=10)

flag=0;

if(flag==0)

{

printf("INVALID NUMBER\n");

return;

}

num2=sum;

while(num2!=0)

{

rem=num2%10;

revnum=(revnum\*10)+rem;

num2=num2/10;

}

if(revnum==sum)

{

printf("%d", sum);

printf(", Number is Palindrome\n");

}

else

{

printf("%d", sum);

printf(", Number is not a Palindrome\n");

}

}