C Program to Check whether an Alphabet is Vowel or Consonant

```
#include <stdio.h>
int main()
{
    char ch;
    bool isVowel = false;

    printf("Enter an alphabet: ");
    scanf("%c",&ch);

    if(ch=='a'||ch=='A'||ch=='e'||ch=='E'||
ch=='i'||ch=='I'
         ||ch=='o'||ch=='O'||ch=='u'||
ch=='U')
    {
     isVowel = true;

    }
    if (isVowel == true)
        printf("%c is a Vowel", ch);
    else
        printf("%c is a Consonant", ch);
    return 0;
}
```

C Program to Count the Number of Vowels, Consonants and so on

In this example, the number of vowels, consonants, digits, and white-spaces in a string entered by the user is counted.

```
#include <stdio.h>
int main() {
    char line[150];
    int vowels, consonant, digit, space;

    vowels = consonant = digit = space = 0;

    printf("Enter a line of string: ");
    fgets(line, sizeof(line), stdin);

    for (int i = 0; line[i] != '\0'; ++i) {
        if (line[i] == 'a' || line[i] == 'e' || line[i] == 'i' ||
            line[i] == 'o' || line[i] == 'u' || line[i] == 'A' ||
            line[i] == 'E' || line[i] == 'I' || line[i] == 'O' ||
            line[i] == 'U') {
            ++vowels;
```

```
        } else if ((line[i] >= 'a' && line[i] <=
'z') || (line[i] >= 'A' && line[i] <= 'Z')) {
            ++consonant;
        } else if (line[i] >= '0' && line[i] <=
'9') {
            ++digit;
        } else if (line[i] == ' ') {
            ++space;
        }
    }

    printf("Vowels: %d", vowels);
    printf("\nConsonants: %d", consonant);
    printf("\nDigits: %d", digit);
    printf("\nWhite spaces: %d", space);
    return 0;
}
```

Leap year or not

```
#include <stdio.h>
int main() {
   int year;
   printf("Enter a year: ");
   scanf("%d", &year);
```

```
    // leap year if perfectly divisible by 400
    if (year % 400 == 0) {
        printf("%d is a leap year.", year);
    }
    // not a leap year if divisible by 100
    // but not divisible by 400
    else if (year % 100 == 0) {
        printf("%d is not a leap year.", year);
    }
    // leap year if not divisible by 100
    // but divisible by 4
    else if (year % 4 == 0) {
        printf("%d is a leap year.", year);
    }
    // all other years are not leap years
    else {
        printf("%d is not a leap year.", year);
    }

    return 0;
}
```

Sum of Natural Numbers Using for Loop

```
#include <stdio.h>
```

```
int main() {
    int n, i, sum = 0;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

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

    printf("Sum = %d", sum);
    return 0;
}
```

C Program to Find GCD of two Numbers

The HCF or GCD of two integers is the largest integer that can exactly divide both numbers (without a remainder).

```
#include <stdio.h>
int main()
{
    int n1, n2, i, gcd;

    printf("Enter two integers: ");
```

```
    scanf("%d %d", &n1, &n2);

    for(i=1; i <= n1 && i <= n2; ++i)
    {
        // Checks if i is factor of both
integers
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }

    printf("G.C.D of %d and %d is %d", n1,
n2, gcd);

    return 0;
}
```

C Program to Find LCM of two Numbers

The LCM of two integers n1 and n2 is the smallest positive integer that is perfectly divisible by both n1 and n2 (without a remainder). For example, the LCM of 72 and 120 is 360.

```
#include <stdio.h>
```

```
int main() {
    int n1, n2, max;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);

    // maximum number between n1 and
n2 is stored in max
    max = (n1 > n2) ? n1 : n2;

    while (1) {
        if (max % n1 == 0 && max % n2 ==
0) {
            printf("The LCM of %d and %d is
%d.", n1, n2, max);
            break;
        }
        ++max;
    }
    return 0;
}
```

check palindrome using while loop

```
/* Program to check if a number is
palindrome or not
 * using while loop
 */
```

```
#include <stdio.h>
int main()
{
   int num, reverse_num=0,
remainder,temp;
   printf("Enter an integer: ");
   scanf("%d", &num);

   /* Here we are generating a new
number (reverse_num)
    * by reversing the digits of original
input number
    */
   temp=num;
   while(temp!=0)
   {
      remainder=temp%10;

reverse_num=reverse_num*10+remainde
r;
      temp/=10;
   }

   /* If the original input number (num) is
equal to
```

```
    * to its reverse (reverse_num) then its
palindrome
    * else it is not.
    */
   if(reverse_num==num)
      printf("%d is a palindrome
number",num);
   else
      printf("%d is not a palindrome
number",num);
   return 0;
}

// Program to calculate the sum of
numbers (10 numbers max)
// If the user enters a negative number,
the loop terminates

#include <stdio.h>

int main() {
   int i;
   double number, sum = 0.0;

   for (i = 1; i <= 10; ++i) {
      printf("Enter n%d: ", i);
      scanf("%lf", &number);
```

```
      // if the user enters a negative
number, break the loop
      if (number < 0.0) {
         break;
      }

      sum += number; // sum = sum +
number;
   }

   printf("Sum = %.2lf", sum);

   return 0;
}
```

Find the Frequency of a Character

```
#include <stdio.h>
int main() {
    char str[1000], ch;
    int count = 0;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Enter a character to find its
frequency: ");
    scanf("%c", &ch);
    for (int i = 0; str[i] != '\0'; ++i) {
```

```
        if (ch == str[i])
            ++count;
    }
    printf("Frequency of %c = %d", ch,
count);
    return 0;
}
```

Output

```
Enter a string: This website is awesome.
Enter a character to find its frequency: e
Frequency of e = 4
```

Check if a character is an alphanumeric character

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);

    if (isalnum(c) == 0)
        printf("%c is not an alphanumeric
character.", c);
```

```
    else
        printf("%c is an alphanumeric
character.", c);

    return 0;
}
```

Program to cyclically rotate an array by one

Given an array, cyclically rotate the array clockwise by one.
Examples:

Input: arr[] = {1, 2, 3, 4, 5}
Output: arr[] = {5, 1, 2, 3, 4}

Following are steps.
1) Store last element in a variable say x.
2) Shift all elements one position ahead.
3) Replace first element of array with x.
Output:

```
Given array is
1 2 3 4 5
Rotated array is
5 1 2 3 4
```

```c
#include <stdio.h>
void rotate(int arr[], int n)
{
  int x = arr[n-1], i;
  for (i = n-1; i > 0; i--)
    arr[i] = arr[i-1];
  arr[0] = x;
}
int main()
{
   int arr[] = {1, 2, 3, 4, 5}, i;
   int n = sizeof(arr)/sizeof(arr[0]);
   printf("Given array is\n");
   for (i = 0; i < n; i++)
      printf("%d ", arr[i]);
   rotate(arr, n);
   printf("\nRotated array is\n");
   for (i = 0; i < n; i++)
      printf("%d ", arr[i]);
   return 0;
}
```

Arrangement and rearrangement of elements of array

Given a sorted array of positive integers, rearrange the array alternately i.e first element should be maximum value, second minimum value, third second max, fourth second min and so on.

Examples:

```
Input  : arr[] = {1, 2, 3, 4, 5, 6, 7}
Output : arr[] = {7, 1, 6, 2, 5, 3, 4}

Input  : arr[] = {1, 2, 3, 4, 5, 6}
Output : arr[] = {6, 1, 5, 2, 4, 3}
```

The idea is use an auxiliary array. We maintain two pointers one to leftmost or smallest element and other to rightmost or largest element. We more both pointers toward each other and alternatively copy elements at these pointers to an auxiliary array. Finally we copy auxiliary array back to original array.

Output:

```
Original Array
1 2 3 4 5 6 7 8 9
```

Modified Array
9 1 8 2 7 3 6 4 5

```
#include <iostream.h>
using namespace std;
void rearrange(int arr[],int n)
{
  int temp[n]; //Auxiliary array to hold
  modified array
  int small=0, large = n-1; //Indexes of
  Smallest and Largest elements from
  remaining array
  int flag=1; //to indicate whether we
  need to copy remaining largest or
  remaining smallest at next position
  for(int i=0;i<n; i++) //Store result in
  temp[]
  {
    if (flag==1)
      temp[i] = arr[large—-];
      flag=0;
    else
      temp[i] = arr[small++];
      flag=1;                          //flag=!
      flag;
  }
```

```
  for(int i=0; i<n; i++) //Copy temp[] to
  arr[]
    arr[i] =temp[i];
}

// Driver program to test above
function
int main()
{
   int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
   int n = sizeof(arr)/sizeof(arr[0]);

   cout << "Original Array";
   for (int i=0; i<n; i++)
      cout << arr[i] << " ";

   rearrange(arr, n);

   cout << "nModified Arrayn";
   for (int i=0; i<n; i++)
      cout << arr[i] << " ";
   return 0;
}
```