

B 101 : The If-Else Statement

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

```
int main()
{
    int var1, var2;
    scanf("%d",&var1);
    scanf("%d",&var2);
    if (var1 < var2)
    {
        printf("Value1 is less than Value2");
    }

    else
    {
        printf("Value1 is not less than Value2");
    }

    return 0;
}
```

B 102 : The If-Else If Statement

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

```
int main()
{
    int var1, var2;
    scanf("%d",&var1);
    scanf("%d",&var2);
    if (var1 > var2)
    {
        printf("value1 is greater");
    }
    else if(var1 == var2){
        printf("both values are equal");
    }
    else{
        printf("value2 is greater");
    }

    return 0;
}
```

B 103 : Check if a Given Number is Even or Odd

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

```
int main()
{
    int var1;
    scanf("%d", &var1);
    if (var1%2 == 0)//Replace the ' _ ' with appropriate conditions and statements
    {
        printf("EVEN");
    }
    else
    {
        printf("ODD");
    }
    return 0;
}
```

B 104 : The For Loop

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

```
int main()
{
    int i;
    for (i = 1; i <= 10; i++)
    {
        printf("%d ",i);
    }
}
```

B 105 : More on For Loops

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

```
int main()
{
    int i, N;
    //input N and print all odd numbers less than N
    //Hint : start a for loop at i = 1, and increment by 2 i.e. i += 2.

    scanf("%d", &N);
    for(i=1; i<N; i+=2)
    {
        printf("%d ",i);
    }
}
```

```
}}
```

B 106 : The While Loop

```
int main()
{
    int var, sum = 0, input_count = 0;
    while (input_count<5)//complete the condition so that you only take 5 numbers as input.
    {
        scanf("%d", &var);
        sum += var;
        input_count ++;
    }
    printf("%d", sum);
    return 0;
}
```

B 107 : Odd or Even

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

```
int main()
{
    int var, input_count = 0;
    //write your code here

    while(input_count < 5){
        scanf("%d",&var);
        if (var%2 == 0){
            printf("even\n");}

        else{
            printf("odd\n");}
        input_count++;
    }
    return 0;
}
```

B 108 : Prime or Not

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

```
int main()
{
    int number, is_prime,i;
    scanf("%d", &number);
```

```

        is_prime=1;
    if(number==1){
        is_prime=0;
    }
    if(number>2){
        for(i=2;i<number;i++){
            if(number%i==0){
                is_prime=0;
                break;
            }
        }
    }
    if (is_prime == 1) printf("yes");
    if (is_prime == 0) printf("no");
    return 0;
}

```

B 109 : The Break Statement

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

```

int is_prime(int num)
{
    if (num < 2) return 0;
    int i;
    for (i = 2; i < num; i++)
    {
        if (num % i == 0) return 0;
    }
    return 1;
}

int main()
{
    int i, num;
    scanf("%d", &num);
    for (i = num + 1; ; i++)
    {
        if (is_prime(i) == 1)//break the loop once youve found the prime
        {
            printf("%d",i);

            break;
        }
    }
}

```

```
return 0;}
```

B 110 : The Continue Statement

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

```
int main()
{
    int i, N;
    scanf("%d", &N);
    for (i = 1; i <= N; i++)
    {
        if (i % 3 == 0)//add a continue statement to move on to the next iteration
        {
            continue;
        }
        printf("%d ", i);
    }
    return 0;
}
```

B 111 : Multiplication Table

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
```

```
    int n;
    cin>>n;
    for(int i=1;i<=10;i++)
    {
        cout<<n*i<<" ";
    }
    return 0;
}
```

B 112 : Goto and Label

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

```
int main()
{
    int num;
```

```

scanf("%d", &num);
while (num > 0)
{
if (num % 2 == 1) //write the goto statement here to skip to the end
goto label;
else
printf("%d ", num);
num = num/2;
}
label : printf("Process Complete");
return 0;
}

```

B 113 : Sum of Digits

```

#include <stdio.h>
//Q13
int main()
{
int n; int sum=0;
scanf("%d",&n);
while(n!=0)
{
sum=sum+n%10;
n=n/10;
}
printf("%d", sum);
return 0;
}

```

B 114 : Nested For Loops

```

#include <stdio.h>
int main()
{
int i, j, n;
scanf("%d", &n);
//Replace the _ with appropriate code to get the desired output.
for (i=1;i<=n;i++)
{
for (j=1;j<=n;j++)
{
printf("*");
}
printf("\n");
}
}

```

B 115 : Nested For Loops - II

```
#include <stdio.h>
int main()
{ int n;
  scanf("%d",&n);
  //Your code here
  for(int i=1;i<=n;i++)
  {
    for(int j=1;j<=i;j++)
    {
      printf("*");
    }
    printf("\n");
  }
  return 0;
}
```

B 116 : The Switch-Case Statement

```
#include <stdio.h>
int main()
{
  char ch;
  scanf("%c", &ch);
  switch(ch)
  {
    case 'a' : printf("vowel"); break;
    case 'e' : printf("vowel"); break;
    //write the cases for i, o, and u.
    case 'i' : printf("vowel"); break;
    case 'o' : printf("vowel"); break;
    case 'u' : printf("vowel"); break;
    default : printf("not vowel"); break;
  }
  return 0;
}
```

B 117 : Grading System

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

```
int main()
{
```

```

int marks;
//char grade;
scanf("%d", &marks);
//write the code to assign the grade
if(marks>=76){
printf("A");
}else if(marks>=51){
printf("B");
}else if(marks>=26){
printf("C");
}else{
printf("D");
}
//printf("%c", grade);
return 0;
}

```

B 202 : FACTORS OF X

#python3

```

t = int(input())

while(t):
    x = int(input())

    for i in range(1,x+1):
        if x%i==0:
            print(i,end=' ')

    print()

    t-=1

```

B 203 : MULTIPLE'S OF 4

```

x = int(input())

if x%4==0:
    print('yes')
else:
    print('no')

```

B 204 : All the factors

```

x = int(input())

```



```

for i in range(1,x+1):
    if x%i==0:
        print(i,end = ' ')

```

B 205 : C D01 - Prime Testing - 1

```

num = int(input())

```

```

if num > 1:
    for i in range(2, int(num/2)+1):
        if (num % i) == 0:
            print("no")
            break
    else:
        print("yes")

```

```

else:
    print("no")

```

C 101 : Arrays

```

#include<stdio.h>
int main()
{
    int i, arr[5]; //creates array arr of size 5
    for (i = 0; i < 5; i++) // loop from i = 0 to i = 4
    {
        scanf("%d", &arr[i]); //replace _ by arr[i] to take input for ith element
    }
    for (i = 0; i < 5; i++)
    {
        printf("%d ", arr[i]); //replace _ to print the ith element.
    }
    return 0;
}

```

C 102 : Printing an Array in Reverse

```

a = list(map(int,input().split(' ')))
print(*a[::-1])

```

C 103 : Using Variable as an Array Size

```

#include<stdio.h>
int main()
{
    int i, N;

```

```

scanf("%d", &N); //acquiring the value of N, the size of array
int arr[N]; //declaring an array of size N.
//input the elements of the array here and then print them in reverse
for(int i=0;i<N;i++)
{
    scanf("%d",&arr[i]);
}
for (i = N-1; i>=0; i--)
{
    printf("%d ", arr[i]);
}

return 0;
}

```

C 104 : Initialization

```

#include<stdio.h>
int main()
{
    int i, a[10] = {0}; //direct initialisation
    for (i = 0; i < 10; i++) scanf("%d", &a[i]);
    //print the odd indexed elements of the array here
    for (i = 0; i < 10; i++)
    {
        if(i%2!=0)
        {
            printf("%d ",a[i]);
        }
    }
    return 0;
}

```

B 501 : Cut the Fruit

```

#include<bits/stdc++.h>
using namespace std;
int main() {
    int t;
    cin>>t;
    if(t==2)
        cout<<"NO"<<endl;
    else if(t%2==0)
        cout<<"YES"<<endl;
    else
        cout<<"NO"<<endl;return 0; }

```

B 502 : Equal Bases

```
c=0
```

```
n=int(input())
```

```
for i in range(2,n+1):
```

```
    if(n%i==0):
```

```
        c+=1
```

```
print(c)
```

B 503 : Trouble with the Number System

```
n=int(input())
```

```
sum1=0
```

```
c=1
```

```
for i in range(n):
```

```
    m=int(input())
```

```
    while(m%10==0):
```

```
        sum1+=1
```

```
        m//=10
```

```
    c*=m
```

```
print(str(c)+"0"*sum1)
```

C 301 : The Elite N

```
#include <cmath>
```

```
#include <cstdio>
```

```
#include <vector>
```

```
#include <iostream>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
int main() {
```

```
    int p,n,count=1,flag=1;
```

```

cin>>p>>n;
int arr[n];
for(int i=0;i<n;i++)
    cin>>arr[i];
int d = p;
for(int i=0;i<n;i++)
{
    if(arr[i]>=p)
    {
        cout<<-1;
        flag = 0;
        break;
    }
    else if(arr[i]<d)
        d-=arr[i];
    else
    {
        d = p;
        count++;
        i--;
    }
}
if(flag)
    cout<<count<<endl;
return 0;
}

```

C 302 : Paneer Love

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main() { int n;  scanf("%d",&n);  int a[n],p[n];  int i;  for(i=0;i<n;i++) {    scanf("%d
%d",&a[i],&p[i]);  }  i=0;  while(i<n-1) {    if(p[i]<p[i+1])    {        p[i+1]=p[i];    }
i++;  }  int min=0;  for(i=0;i<n;i++) {    min=min+a[i]*p[i];    }  printf("%d",min);
return 0;}

```

E 101 : Introduction to Pointers

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h> //this library contains the function malloc.
int main()
{
    int *var = (int*) malloc(sizeof(int)); //allocating memory to a new int pointer

```

```

scanf("%d", var); //store the input to var. recall that a pointer is address itself so you
//do not need to use the ampersand '&' (address of) sign.
printf("%d", *var); //print the variable var here. again, recall that you need to use the
//indirection/dereference operator '*' to access a pointers value.
return 0;
}

```

E 102 : Array Using Pointers

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h> //this library contains the function malloc.
int main()
{
    int i, N, sum = 0;
    scanf("%d", &N);
    int *arr = (int*) malloc(N*sizeof(int)); //allocate memory to a new int pointer
    for (i = 0; i < N; i++) scanf("%d", (arr + i)); //can also write &arr[i];
    for (i = 0; i < N; i++) sum += *(arr + i); //can also write arr[i];
    printf("%d", sum);
    return 0;
}

```

E 103 : Pointer to an Array

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void replace(double* pointer, int arr_size)
{
    for(int i=0;i<5;i++) pointer[i] = sqrt(pointer[i]);
}
int main()
{
    int i;
    double arr[5];
    for (i = 0; i < 5; i++) scanf("%lf", &arr[i]);
    double* p = arr; //can also use double* p = &arr[0];
    replace(p, 5); //changes to p will reflect in array arr
    for (i = 0; i < 5; i++) printf("%.4lf ", arr[i]);
    return 0;
}

```

E 104 : Memory Allocation Functions

```

#include<stdio.h>

```

```

#include<string.h>
#include<stdlib.h>
#include<ctype.h>
char* newstr(char* str)
{int i, n = strlen(str);
  char *res=(char*)malloc(n*sizeof(char)); //use malloc to declare this character array

  for (i = 0; i < n; i++)
  {
    if (islower(str[i])) res[i] = toupper(str[i]);
    else res[i] = tolower(str[i]);
  }
  res[n] = '\0';
  return res;
}
int main()
{
  char str[50];
  gets(str);
  char* new_str = newstr(str);
  puts(new_str);
  return 0;
}

```

E 105 : Resizing an Array

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
  int* arr = (int*) malloc(sizeof(int) * 3);
  int i, n;
  //input N and then use realloc to change size of array arr.
  scanf("%d",&n);
  realloc(arr, n*sizeof(int));
  for( i=0;i<n;i++) scanf("%d",&arr[i]);
  for( i=n-1;i>=0;i--) printf("%d ",arr[i]);
  return 0;
}

```

F 101 : Introduction to Strings

```

#include<stdio.h>
#include<string.h>
int main()
{

```

```

char str[100] = "Hello World!";
printf("%s", str);
return 0;
}

```

F 102 : Input a String

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[50];
    //input string here using scanf
    scanf("%s",str);
    //print string here using printf
    printf("%s",str);
    return 0;
}

```

F 103 : Gets and Puts

```

#include<stdio.h>
#include<string.h>
int main()
{
    char str[50];
    //input string here using gets
    gets(str);
    //print string here using puts
    printf("%s",str);
    return 0;
}

```

F 104 : Find the Length of the String

```

#include <stdio.h>

int main() {
    char str[100];
    unsigned short count = 0;

    scanf("%[^\\n]", str);

    while (str[count] != '\\0')
        count++;
}

```

```

printf("%u", count);

return 0;
}

```

F 105 : Count Words

```

#include <stdio.h>
#include <string.h>

int main()
{
    char s[200];
    int count = 0, i;

    scanf("%[^\n]s", s);
    for (i = 0; s[i] != '\0'; i++)
    {
        if (s[i] == ' ' && s[i+1] != ' ')
            count++;
    }
    printf("%d\n", count + 1);
}

```

S 101 : Vowels on Strike

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    char s[1000];
    fgets(s, 1000, stdin);
    for(int i=0;i<strlen(s);i++) if(!(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u' ||
s[i]=='A' || s[i]=='E' || s[i]=='I' || s[i]=='O' || s[i]=='U')) printf("%c", s[i]);
    return 0;
}

```

Task 4.10 : Palindrome

```

#include <stdio.h>
#include <string.h>

int main(){
    char string1[30];
    int i, length;

```



```

int flag = 0;

scanf("%s", string1);

length = strlen(string1);

for(i=0;i < length ;i++){
    if(string1[i] != string1[length-i-1]){
        flag = 1;
        break;
    }
}

if (flag) {
    printf("0");
}
else {
    printf("1");
}
return 0;
}

```

Task 4.10 : Leap Years

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
char** leapeyears (int arr[], int size)
{
    //declare dynamically an array of strings result
    char **result = (char**)malloc(size*sizeof(char*));
    for(int i=0;i<size;i++){
        result[i] = (char*)malloc(size*sizeof(char));
    }
    char yes[] = "yes";
    char no[] = "no";

    for (int i = 0; i < size; i++)
    {
        if (arr[i]%4) strcpy(result[i], no);
        else strcpy(result[i], yes); //replace the '_' with proper condition
    }
    return result;
}
int main()

```

```

{
    int i, N;
    scanf("%d", &N);
    int arr[N];
    for (i = 0; i < N; i++) scanf("%d", &arr[i]);
    char** ans = leapyears(arr, N);
    for (i = 0; i < N; i++) puts(ans[i]);
    return 0;
}

```

C 501 : Array as a Hill

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main() {
    int n;
    scanf("%d",&n);
    int a[n];
    for (int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    int i=0;

    while (i<n-1)

    {

        if(a[i]<a[i+1])

            i++;

        else break;

    }

    while (i<n-1)

    {

        if(a[i]==a[i+1])

```

```

        i++;

    else break;

}

while (i<n-1)

{

    if(a[i]>a[i+1])

        i++;

        else break;

}

if(i==n-1)

    printf("yes");

else

    printf("no");

return 0;

}

```

Z 405 : Concatenate Two Strings

```

#include<stdio.h>
void strcat(char* s1, char* s2)
{
}
int main()
{
    char str1[50], str2[50];
    gets(str1);
    gets(str2);
    strcat(str1, str2);
    printf("%s%s",str1,str2);
    //puts(str1);
    return 0;
}

```

Z 407 : The Strcmp Function

```
#include<stdio.h>

int strcmp(char* s1,char* s2){
    int sum1=0,sum2=0;
    for(int i=0;s1[i]!='\0' && s2[i]!='\0';i++){
        if(s1[i]!=s2[i]){
            return (int)s1[i]-(int)s2[i];
        }
    }
    return 0;
}

int main(){
    char s1[50],s2[50];
    gets(s1);
    gets(s2);
    printf("%d",strcmp(s1,s2));
    return 0;
}
```

Z 407 : The StrCpy Function

```
#include<stdio.h>
void strcpy(char* s1, char* s2)
{
    //your code here
}
int main()
{
    char str1[50], str2[50];
    gets(str1);
    strcpy(str2, str1);
    puts(str1);
    puts(str2);
    return 0;
}
```

F 501 : String Sans Accent

```
x = input()
vow = ['a','e','i','o','u']
for i in x:
    if i not in vow:
```

```
print(i,end=' ')
```

F 503 Reverse Vowels

```
x = input()
vow = ['a','e','i','o','u']
vs = [i for i in x if i in vow]
vs = vs[::-1]
c=0
for i in x:
    if i in vs:
        print(vs[c],end = " ")
        c+=1
    else:
        print(i,end = " ")
```

H 101 : Basics of Functions

```
#include<stdio.h>
//write your function print() here
void print()
{
    printf("Hello World!");
}
int main()
{
    print();
    return 0;
}
```

H 102 : Function Prototype

```
#include<stdio.h>
//add function prototype here

int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    int sum = add(a, b);
    printf("%d", sum);
    return 0;
}
int add(int var1, int var2)
{
    return (var1 + var2);
}
```

H 103 : Compare Function

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

```
int compare(int var1,int var2){
```

```
    if(var1>var2)
```

```
        return 1;
```

```
    else if(var2>var1)
```

```
        return -1;
```

```
    return 0;
```

```
}
```

```
int main()
```

```
{
```

```
    int val1, val2;
```

```
    scanf("%d %d", &val1, &val2);
```

```
    int cmp;
```

```
    cmp = compare(val1, val2);
```

```
    if (cmp == 1) printf("Value 1 is greater");
```

```
    else if (cmp == -1) printf("Value 2 is greater");
```

```
    else printf("Both Values are equal");
```

```
    return 0;
```

```
}
```

H 104 : Call by Value and Call by Reference

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

```
#include<math.h>
```

```
void power(int*, int);
```

```
int main()
```

```
{
```

```
    int res, val;
```

```
    scanf("%d", &val);
```

```
    res = pow(2, val);
```

```
    printf("%d", res);
```

```
    return 0;
```

```
}
```

H 105 : The Swap Function

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

```
void swap(int*, int*);
```

```
int main()
```

```
{
```

```

    int v1, v2;
    scanf("%d%d", &v1, &v2);
    swap(&v1, &v2);
    printf("%d %d", v1, v2);
    return 0;
}
//implement the function swap here
void swap(int *x,int *y){
    int t = *x;
    *x = *y;
    *y = t;
}

```

H 106 : Standard Library Functions

```

#include<stdio.h>
#include<math.h>
double hypotenuse(double base, double perp)
{
    double h;
    h=sqrt((base*base)+(perp*perp));
    return h;
    //complete this function
}
int main()
{
    double A, B, C;
    scanf("%lf%lf", &A, &B);
    C = hypotenuse(A, B);
    printf("%.6lf", C);
    return 0;
}

```

H 107 : Character Type Functions

```

#include<stdio.h>
#include<ctype.h>
int main()
{
    char ch;
    scanf("%c", &ch);
    //your code here
    if((int)ch<95)
        printf("%c",tolower(ch));
    else

```

```

    printf("%c",toupper(ch));
    return 0;
}

```

I 101 : Recursive Functions

void func(int n)//modify this function so it prints numbers in reverse

```

{
    if (n <1)
        return; //this is the base case.
    printf("%d ", n);
    func(n - 1);
}

```

//Hint : you can change the order of statements in the function to achieve your goal.

```

int main()
{
    func(5);
    return 0;
}

```

I 102 : More Uses of Recursion

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

```
long int multiplyNumbers(int n);
```

```

int main() {
    int n;
    scanf("%d",&n);
    printf("%d ",multiplyNumbers(n));
    return 0;
}

```

```

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}

```

I 103 : The Fibonacci Series

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

```
int fibonacci(int N)
```

```

{
    if (N <= 1)
        return N;
    return fibonacci(N-1) + fibonacci(N-2);
}

```



```

int main()
{
int N;
scanf("%d", &N);
printf("%d", fibonacci(N));
return 0;
}

```

I 301 : Natural Numbers Recursively

```

#include<stdio.h>
void display(int);
int main()
{
int limit;
scanf("%d", &limit);
display(limit);
return 0;
}
void display(int num)
{
if(num)
display(num-1);
else
return;
printf("%d ", num);
}

```

I 302 : Fibonacci Number Using Recursion

```

def fib(n):
    if n==0 or n==1:
        return n
    return fib(n-1)+ fib(n-2)
x = int(input())
print(fib(x))

```

I 303 : Factorial Using Recursive Function

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {

```

```

int n;

scanf("%d",&n);
printf("%ld", multiplyNumbers(n));
return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}

```

Z 321 Add Two Values

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main() {
    float a,b;
    scanf("%f%f",&a,&b);
    long long int c=a+b;
    if(c<(a+b))

        printf("%.2f",a+b);

    else

        printf("%lld",c);

    return 0;
}

```

I M16 - Exclusive Or

```
x=input()
y=input()
if len(x) > len(y):
    y=y.zfill(len(x))
else:
    x = x.zfill(len(y))

for i in range(len(x)):
    if x[i] == y[i] :
        print('0',end="")
    else:
        print('1',end="")
```

C D09 - To and Fro

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main() {
    int t;
    scanf("%d",&t);
    while(t-->0)
    {
        int f,b,t,d,td=0;
        scanf("%d %d %d %d", &f,&b,&t,&d);
        while(b<d)
        {
            td=td+b+f;
            d=d-(b-f);
        }

        td=td+d;

        printf("%d\n", td*t);
    }

    return 0;
}
```

Mars Exploration

```
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <assert.h>
#include <limits.h>
#include <stdbool.h>

int main(){
    char* S = (char *)malloc(10240 * sizeof(char));
    scanf("%s",S);
    int i;
    int count=0;
    for(i=0;S[i]!='\0';i+=3){
        if(S[i]!='S'){
            count++;
        }
        if(S[i+1]!='O'){
            count++;
        }
        if(S[i+2]!='S'){
            count++;
        }
    }
    printf("%d",count);
    return 0;
}
```

Caesar Cipher

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int n,i,j,k;
    char ar[101];
    unsigned char x;
    scanf("%d",&n);
    scanf("%s",ar);
    scanf("%d",&k);
    for(i=0;i<n;i++)
    {
```

```
x=ar[i];
if(x>=97 && x<=122)
{
    x=x+(k%26);
    if(x>122)
    {
        x=96+(x-122);
    }
    ar[i]=x;
}
else if(x>=65 && x<=90)
{
    x=x+(k%26);
    if(x>90)
    {
        x=64+(x-90);
    }
    ar[i]=x;
}
}
printf("%s",ar);
return 0;
}
```

