Programming assignment

By : Reem Mahmoud El-sayed

Strings

C Program to Find the Frequency of Characters in a String :

```c

#include <stdio.h>
int main()
{
    char str[1000], ch;
    int i, frequency = 0;
    printf("Enter a string: ");
    gets(str);
    printf("Enter a character to find the frequency: ");
    scanf("%c",&ch);
    for(i = 0; str[i] != '\0'; ++i)
    {     if(ch == str[i])
            ++frequency;    }
    printf("Frequency of %c = %d", ch, frequency);
    return 0;
}


```

C Program to Remove all Characters in a String Except Alphabet :

```c
#include<stdio.h>

int main()
{
    char line[150];
    int i, j;
    printf("Enter a string: ");
    gets(line);

    for(i = 0; line[i] != '\0'; ++i)
    {
        while (!( (line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A' && line[i] <= 'Z') || line[i] == '\0') )
        {
            for(j = i; line[j] != '\0'; ++j)
            {
                line[j] = line[j+1];
            }
            line[j] = '\0';
        }
    }
    printf("Output String: ");
    puts(line);
    return 0;
}
```

C Program to Check if the Substring is present in the given String :

```c
#include<stdio.h>
void main()
{
    char str[80], search[10];
    int count1 = 0, count2 = 0, i, j, flag;
    printf("Enter a string:");
    gets(str);
    printf("Enter search substring:");
    gets(search);
    while (str[count1] != '\0')
        count1++;
    while (search[count2] != '\0')
        count2++;
    for (i = 0; i <= count1 - count2; i++)
    {
        for (j = i; j < i + count2; j++)
        {
            flag = 1;
            if (str[j] != search[j - i])
            {
                flag = 0;break;}
        }
        if (flag == 1)
            break;
    }
    if (flag == 1)
        printf("SEARCH SUCCESSFUL!");
    else
        printf("SEARCH UNSUCCESSFUL!");
}
```

C Program to Replace Lowercase Characters by Uppercase & Vice-Versa :

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

void main()
{
    char sentence[100];
    int count, ch, i;

    printf("Enter a sentence \n");
    for (i = 0; (sentence[i] = getchar()) != '\n'; i++)
    {
        ;
    }
    sentence[i] = '\0';
    count = i;
    printf("The given sentence is    : %s", sentence);
    printf("\n Case changed sentence is: ");
    for (i = 0; i < count; i++)
    {
        ch = islower(sentence[i])? toupper(sentence[i]) :
tolower(sentence[i]);
        putchar(ch);
    }
}
```

C Program to Remove given Word from a String:

```c
#include <string.h>
void main()
{
    int i, j = 0, k = 0, count = 0;
    char str[100], key[20];
    char str1[10][20];
    printf("enter string:");
    scanf("%[^\n]s",str);
    for (i = 0; str[i]!= '\0'; i++)
    {if (str[i]==' ')
        {str1[k][j] = '\0';
            k++;
            j = 0;}
        else
        {str1[k][j] = str[i];
            j++;}}
    str1[k][j] = '\0';
    printf("enter key:");
    scanf("%s", key);
    for (i = 0;i < k + 1; i++)
    {if (strcmp(str1[i], key) == 0)
        {for (j = i; j < k + 1; j++)
            strcpy(str1[j], str1[j + 1]);
            k--;}}
    for (i = 0;i < k + 1; i++)
    {
        printf("%s ", str1[i]);
    }
}
```

C Program to Delete All Repeated Words in String:

```c
#include <stdio.h>
#include <string.h>
void main()
{
    char a[100], b[20][20];
    int i, j = 0, k = 0, n, m;
    printf("enter the string\n");
    scanf("%[^\n]s", a);
    for (i = 0;a[i] != '\0';i++)
    {if (a[i] == ' ')
        {b[k][j] = '\0';
            k++;
            j = 0;}
        else
        {b[k][j] = a[i];
            j++;}}
    b[k][j] = '\0';
    for (i = 0;i <= k;i++)
    {for (j = i + 1;j <= k;j++)
        {if (strcmp(b[i], b[j]) == 0)
            {for (m = j;m <= k;m++)
                    strcpy(b[m], b[m + 1]);
                k--;}}}
    for (n = 0;n <= k;n++)
    {
        printf("%s\n", b[n]);
    }
}
```

C Program to Count the Number of Vowels & Consonants in a Sentence:

```c
#include <stdio.h>
void main()
{
    char sentence[80];
    int i, vowels = 0, consonants = 0, special = 0;
    printf("Enter a sentence \n");
    gets(sentence);
    for (i = 0; sentence[i] != '\0'; i++)
    {
        if ((sentence[i] == 'a' || sentence[i] == 'e' || sentence[i] ==
        'i' || sentence[i] == 'o' || sentence[i] == 'u') ||
        (sentence[i] == 'A' || sentence[i] == 'E' || sentence[i] ==
        'I' || sentence[i] == 'O' || sentence[i] == 'U'))
        {
            vowels = vowels + 1;
        }
        else
        {
            consonants = consonants + 1;
        }
        if (sentence[i] =='t' ||sentence[i] =='\0' || sentence[i] ==' ')
        {
            special = special + 1;
        }
    }
    consonants = consonants - special;
    printf("No. of vowels in %s = %d\n", sentence, vowels);
    printf("No. of consonants in %s = %d\n", sentence, consonants);
}
```

C Program to Remove all Characters in Second String which are present in First String:

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define CHAR_SIZE 26
void alphacheck(char *, int []);
void create(char [], char [], int[]);
int main(){char str1[50], str2[50];
    int a1[CHAR_SIZE] = {0};
    char str2_rem[50];
    printf("Enter string1: ");
    scanf("%s", str1);
    printf("Enter string2: ");
    scanf("%s", str2);
    alphacheck(str1, a1);
    create(str2_rem, str2, a1);
    printf("On removing characters from second string we get: %s\n", str2_rem)
    return 0;}
void alphacheck(char *str, int a[])
{int i, index;
    for (i = 0; i < strlen(str); i++)
    {str[i] = tolower(str[i]);
        index = str[i] - 'a';
        if (!a[index])
        {a[index] = 1;}}
    printf("\n");}
void create(char str_rem[], char str[], int list[])
{int i, j = 0, index;
for (i = 0; i < strlen(str); i++)
    {index = str[i] - 'a';
        if (!list[index])
        {str_rem[j++] = str[i];}}
    str_rem[j] = '\0';}
```

C anagram programming :

```c
#include <stdio.h>
int check_anagram(char [], char []);
int main()
{char a[100], b[100];
   int flag;
   printf("Enter first string\n");
   gets(a);
   printf("Enter second string\n");
   gets(b);
   flag = check_anagram(a, b);
   if (flag == 1)
      printf("\"%s\" and \"%s\" are anagrams.\n", a, b);
   else
      printf("\"%s\" and \"%s\" are not anagrams.\n", a, b);
   return 0;}
int check_anagram(char a[], char b[])
{int first[26] = {0}, second[26] = {0}, c = 0;
   while (a[c] != '\0')
   {first[a[c]-'a']++;
      c++;}
   c = 0;
   while (b[c] != '\0')
   {second[b[c]-'a']++;
      c++;}
   for (c = 0; c < 26; c++)
   {
      if (first[c] != second[c])
         return 0;
   }

   return 1;
}
```

Write a c program which prints initial of any name:

```c
#include<stdio.h>
int main(){
    char str[20];
    int i=0;
    printf("Enter a string: ");
    gets(str);
    printf("%c",*str);
    while(str[i]!='\0'){
        if(str[i]==' '){
            i++;
            printf("%c",*(str+i));
        }
        i++;
    }
    return 0;
}
```

Pointers

C Program to Find Transpose of a Matrix:

```c
#include <stdio.h>
int main()
{int a[10][10], transpose[10][10], r, c, i, j;
    printf("Enter rows and columns of matrix: ");
    scanf("%d %d", &r, &c);
    printf("\nEnter elements of matrix:\n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {printf("Enter element a%d%d: ",i+1, j+1);
            scanf("%d", &a[i][j]);}
    printf("\nEntered Matrix: \n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {printf("%d  ", a[i][j]);
            if (j == c-1)
                printf("\n\n");}
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {transpose[j][i] = a[i][j];}

    printf("\nTranspose of Matrix:\n");
    for(i=0; i<c; ++i)
        for(j=0; j<r; ++j)
        {
            printf("%d  ",transpose[i][j]);
            if(j==r-1)
                printf("\n\n");
        }

    return 0;
}
```

C Program to Access Elements of an Array Using Pointer:

```c
#include <stdio.h>

int main()
{
    int data[5], i;
    printf("Enter elements: ");

    for(i = 0; i < 5; ++i)
      scanf("%d", data + i);

    printf("You entered: \n");
    for(i = 0; i < 5; ++i)
       printf("%d\n", *(data + i));

    return 0;
}
```

C Program Swap Numbers in Cyclic Order Using Call by Reference:

```c
#include<stdio.h>
void cyclicSwap(int *a,int *b,int *c);

int main()
{
    int a, b, c;

    printf("Enter a, b and c respectively: ");
    scanf("%d %d %d",&a,&b,&c);

    printf("Value before swapping:\n");
    printf("a = %d \nb = %d \nc = %d\n",a,b,c);

    cyclicSwap(&a, &b, &c);

    printf("Value after swapping:\n");
    printf("a = %d \nb = %d \nc = %d",a, b, c);

    return 0;
}
void cyclicSwap(int *a,int *b,int *c)
{

    int temp;
    temp = *b;
    *b = *a;
    *a = *c;
    *c = temp;
}
```

CONCATENATION OF TWO STRINGS USING POINTER IN C PROGRAM:

```c
#include<stdio.h>
int main(){
  int i=0,j=0;
  char *str1,*str2,*str3;
  puts("Enter first string");
  gets(str1);
  puts("Enter second string");
  gets(str2);
  printf("Before concatenation the strings are\n");
  puts(str1);
  puts(str2);
  while(*str1){
      str3[i++]=*str1++;
  }
  while(*str2){
      str3[i++]=*str2++;
  }
  str3[i]='\0';
  printf("After concatenation the strings are\n");
  puts(str3);
  return 0;
}
```

C Program to Find Length of the String using Pointer:

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

int string_ln(char*);

void main() {
   char str[20];
   int length;

   printf("Enter any string : ");
   gets(str);

   length = string_ln(str);
   printf("The length of the given string %s is : %d", str, length);
   getch();
}

int string_ln(char*p)
{
   int count = 0;
   while (*p != '\0') {
      count++;
      p++;
   }
   return count;
}
```

C program to Calculate Area of Circle using Pointers:

```c
#include<stdio.h>

void func(int r, float *a, float *p )
{
    *a = 3.14 * r * r ;
    *p = 2 * 3.14 * r ;
}

void main( )
{
   int radius ;
   float area, perimeter ;
   printf ( "nEnter radius of a circle " ) ;
   scanf ( "%d", &radius ) ;

   func( radius, &area, &perimeter ) ;

   printf ( "Area = %f", area ) ;
   printf ( "nPerimeter = %f", perimeter ) ;
}
```

C program to reverse a string using pointers:

```c
#include<stdio.h>
int main(){
    char str[50];
    char rev[50];
    char *sptr = str;
    char *rptr = rev;
    int i=-1;

    printf("Enter any string : ");
    scanf("%s",str);

    while(*sptr){
     sptr++;
     i++;
    }

    while(i>=0){
     sptr--;
     *rptr = *sptr;
     rptr++;
     --i;
    }

    *rptr='\0';

    printf("Reverse of string is : %s",rev);

    return 0;
}
```

Program to print addition of two matrices using pointers:

```c
#include<stdio.h>
#include<conio.h>
int a[5][5],b[5][5],row,col;
void add(int(*)[5]);
int main()
{int c[5][5],i,j;
    printf("Enter row : ");
    scanf("%d",&row);
    printf("Enter column : ");
    scanf("%d",&col);
    printf("Enter matrix A :\n");
    for(i=0;i<row;i++)
    {for(j=0;j<col;j++)
        {scanf("%d",&a[i][j]);}}
    printf("Enter matrix B :\n");
    for(i=0;i<row;i++)
    {for(j=0;j<col;j++)
        {scanf("%d",&b[i][j]);}}
    add(c);
    printf("Addition :\n");
    for(i=0;i<row;i++)
    {for(j=0;j<col;j++)
        {printf("%d\t",c[i][j]);}
        printf("\n");}
    getch();
    return 0;}
void add(int c[5][5])
{int i,j;
for(i=0;i<row;i++)
    {for(j=0;j<col;j++)
        {c[i][j]=a[i][j]+b[i][j];}}}
```

C program-To multiply two matrices using pointers:

```c
#include<stdio.h>
 #include<stdlib.h>
 int main(void)
{ int a[10][10],b[10][10],c[10][10],n=0,m=0,i=0,j=0,p=0,q=0,k=0;
    int *pt,*pt1,*pt2;
 printf("Enter size of 1st 2d array : ");
 scanf("%d %d",&n,&m);
    for(i=0;i<n;i++)
 {for(j=0;j<m;j++)
    {printf("Enter element no. %d %d :",i,j);
 scanf("%d",&a[i][j]);}}
 printf("Enter size of 2nd 2d array : ");
    scanf("%d %d",&p,&q);
 for(i=0;i<p;i++)
 {for(j=0;j<q;j++)
    {printf("Enter element no. %d %d :",i,j);
 scanf("%d",&b[i][j]);}}
 if(m!=p)
{printf("Multiplication cannot be done\n");
 exit (0);}
 pt=&a[0][0];
 pt1=&b[0][0];
pt2=&c[0][0];
 for(i=0;i<n;i++)
{for(k=0;k<q;k++)
{*(pt2+(i*10+k))=0;
    for(j=0;j<m;j++)
 {*(pt2+(i*10+k))+=*(pt+(i*10+j))**(pt1+(j*10+k));}}}
 for(i=0;i<n;i++)
{for(j=0;j<q;j++)
{printf("%d ",c[i][j]);}
 printf("\n");}
 return 0;}
```

C Program Sort a List of Strings using Pointers:

```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{char *x[20];
int i,n=0;
void reorder(int n,char *x[]);
 printf("Enter no. of String : ");
 scanf("%d",&n);
printf("\n");
for(i=0;i<n;i++)
{printf("Enter the Strings %d : ",i+1);
x[i]=(char *)malloc(20*sizeof(char));
scanf("%s",x[i]);}
reorder(n,x);
printf("\nreorder list is :  \n");
for(i=0;i<n;i++)
{printf("%d %s\n",i+1,x[i]);}
getch();}
void reorder(int n,char *x[])
{int i,j;
char t[20];
for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)
if(strcmp(x[i],x[j])>0)
{strcpy(t,x[j]);
strcpy(x[j],x[i]);
strcpy(x[i],t);}
return;}
```

Recursion

C Program to Find G.C.D Using Recursion:

```c
#include <stdio.h>
int hcf(int n1, int n2);
int main()
{
   int n1, n2;
   printf("Enter two positive integers: ");
   scanf("%d %d", &n1, &n2);

   printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1,n2));
   return 0;
}

int hcf(int n1, int n2)
{
    if (n2 != 0)
       return hcf(n2, n1%n2);
    else
       return n1;
}
```

Multiplication using recursion:

```c
#include<stdio.h>
int multiply(int,int);
int main(){
    int a,b,product;
    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);
    product = multiply(a,b);
printf("Multiplication of two integers is %d",product);
    return 0;}
int multiply(int a,int b){
    static int product=0,i=0;
    if(i < a){
         product = product + b;
         i++;
         multiply(a,b);}
    return product;
}
```

Decimal to binary conversion in c using recursion:

```c
#include<stdio.h>
long toBinary(int);
int main(){
long binaryNo;
int decimalNo;
printf("Enter any decimal number: ");
scanf("%d",&decimalNo);
binaryNo = toBinary(decimalNo);
    printf("Binary value is: %ld",binaryNo);
return 0;}
long toBinary(int decimalNo){
static long binaryNo,remainder,factor = 1;
if(decimalNo != 0){
remainder = decimalNo % 2;
binaryNo = binaryNo + remainder * factor;
factor = factor * 10;
toBinary(decimalNo / 2);}
return binaryNo;}
```

C program for binary search using recursion:

```c
#include<stdio.h>
int main(){
int a[10],i,n,m,c,l,u;
printf("Enter the size of an array: ");
scanf("%d",&n);
printf("Enter the elements of the array: " );
for(i=0;i<n;i++){
scanf("%d",&a[i]);}
printf("Enter the number to be search: ");
scanf("%d",&m);
l=0,u=n-1;
c=binary(a,n,m,l,u);
if(c==0)
printf("Number is not found.");
else
printf("Number is found.");
return 0;}
int binary(int a[],int n,int m,int l,int u){
int mid,c=0;
if(l<=u){
mid=(l+u)/2;
if(m==a[mid]){
c=1;}
else if(m<a[mid]){
return binary(a,n,m,l,mid-1);}
else
return binary(a,n,m,mid+1,u);}
else
return c;}
```

Reverse a string using recursion in c:

```c
#include<stdio.h>
#define MAX 100
char* getReverse(char[]);
int main(){
    char str[MAX],*rev;
    printf("Enter  any string: ");
    scanf("%s",str);
    rev = getReverse(str);
    printf("Reversed string is: %s",rev);
    return 0;}
char* getReverse(char str[]){
    static int i=0;
    static char rev[MAX];
    if(*str){
        getReverse(str+1);
        rev[i++] = *str;}
    return rev;}
```

C Program to Solve Tower-of-Hanoi Problem using Recursion:

```c
#include <stdio.h>
void towers(int, char, char, char);
int main()
{int num;
printf("Enter the number of disks : ");
scanf("%d", &num);
printf("The sequence of moves involved in the Tower of Hanoi are :\n");
towers(num, 'A', 'C', 'B');
 return 0;}
void towers(int num, char frompeg, char topeg, char auxpeg)
{if (num == 1)
{printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);
return;}
towers(num - 1, frompeg, auxpeg, topeg);
printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
towers(num - 1, auxpeg, topeg, frompeg);}
```

Find Sum of Digits of the Number using Recursive Function :

```c
#include <stdio.h>
int sum (int a);
int main()
{int num, result;
printf("Enter the number: ");
scanf("%d", &num);
result = sum(num);
printf("Sum of digits in %d is %d\n", num, result);
return 0;}
int sum (int num)
{if (num != 0)
{return (num % 10 + sum (num / 10));}
    else
    {return 0;}}
```

Prime number program in c using recursion:

```c
#include<stdio.h>
int isPrime(int,int);
int main(){
int num,prime;
printf("Enter a positive number: ");
scanf("%d",&num);
prime = isPrime(num,num/2);
if(prime==1)
printf("%d is a prime number",num);
   else
printf("%d is not a prime number",num);
return 0;}
int isPrime(int num,int i){
if(i==1){
return 1;}
else{if(num%i==0)
return 0;
else
isPrime(num,i-1);}}
```

LCM using recursion in c:

```c
#include<stdio.h>
int lcm(int,int);
int main(){
int a,b,l;
printf("Enter any two positive integers ");
scanf("%d%d",&a,&b);
if(a>b)
l = lcm(a,b);
else
l = lcm(b,a);
printf("LCM of two integers is %d",l);
return 0;}
int lcm(int a,int b){
static int temp = 1;
if(temp % b == 0 && temp % a == 0)
return temp;
temp++;
lcm(a,b);
return temp;
}
```

C Program to find HCF of a given Number using Recursion:

```c
#include <stdio.h>
int hcf(int, int);
int main()
{int a, b, result;
printf("Enter the two numbers to find their HCF: ");
scanf("%d%d", &a, &b);
result = hcf(a, b);
printf("The HCF of %d and %d is %d.\n", a, b, result);}
int hcf(int a, int b)
{while (a != b)
    {if (a > b)
    {return hcf(a - b, b);}
        else
        {return hcf(a, b - a);}}
    return a;}
```

Functions

C Program to Display Prime Numbers Between Intervals Using Function:

```c
#include <stdio.h>
int checkPrimeNumber(int n);
int main()
{int n1, n2, i, flag;
printf("Enter two positive integers: ");
scanf("%d %d", &n1, &n2);
printf("Prime numbers between %d and %d are: ", n1, n2);
for(i=n1+1; i<n2; ++i)
    {flag = checkPrimeNumber(i);
if(flag == 1)
printf("%d ",i);}
    return 0;}
int checkPrimeNumber(int n)
{int j, flag = 1;
for(j=2; j <= n/2; ++j)
    {if (n%j == 0)
    {flag =0;
        break;}}
    return flag;}
```

C Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers:

```c
#include <stdio.h>
int checkPrime(int n);
int main()
{int n, i, flag = 0;
printf("Enter a positive integer: ");
scanf("%d", &n);
for(i=2;i<=n/2; ++i)
    {if (checkPrime(i) == 1)
    {if (checkPrime(n-i) == 1)
        {printf("%d = %d + %d\n", n, i, n - i);
        flag = 1;}}}
        if (flag == 0)
printf("%d cannot be expressed as the sum of two prime numbers.", n);
return 0;}
int checkPrime(int n)
{int i, isPrime = 1;
for(i = 2; i <= n/2; ++i)
    {if(n % i == 0)
    {isPrime = 0;
    break;}}
return isPrime;
}
```

C program to generate and print Armstrong numbers using function:

```c
#include <stdio.h>
int check_armstrong(int);
int power(int, int);
int main () {
   int c, a, b;
   printf("Input two integers\n");
   scanf("%d%d", &a, &b);
   for (c = a; c <= b; c++) {
      if (check_armstrong(c) == 1)
         printf("%d\n", c);}
return 0;}
int check_armstrong(int n) {
   long long sum = 0, temp;
   int remainder, digits = 0;
   temp = n;
   while (temp != 0) {
      digits++;
      temp = temp/10;}
   temp = n;
while (temp != 0) {
remainder = temp%10;
sum = sum + power(remainder, digits);
temp = temp/10;}
   if (n == sum)
return 1;
else
return 0;}
int  power(int n, int r) {
   int c, p = 1;
   for (c = 1; c <= r; c++)
      p = p*n;
   return p;}
```

c program for find year is leap year or not using user define function leap:

```c
#include<stdio.h>
int leap(int );
int main()
{int year;
 printf("Enter any year : ");
 scanf("%d", &year);
 if(leap(year))
   printf("\n%d is leap year",year);
 else
   printf("\n%d is not leap year",year);
 return 0;}
int leap(int y)
{if((y%400==0 && y%100==0)||(y%4==0))
    return 1;
 else
    return 0;}
```

C program to find nCr and nPr using function:

```c
#include <stdio.h>
long factorial(int);
long find_ncr(int, int);
long find_npr(int, int);
int main()
{int n, r;
long ncr, npr;
printf("Enter the value of n and r\n");
scanf("%d%d",&n,&r);
ncr = find_ncr(n, r);
npr = find_npr(n, r);
printf("%dC%d = %ld\n", n, r, ncr);
printf("%dP%d = %ld\n", n, r, npr);
return 0;}
long find_ncr(int n, int r)
{long result;
result = factorial(n)/(factorial(r)*factorial(n-r));
return result;}
long find_npr(int n, int r)
{long result;
result = factorial(n)/factorial(n-r);
   return result;}
long factorial(int n)
{int c;
long result = 1;
   for (c = 1; c <= n; c++)
    result = result*c;
   return result;}
```

C Program for Pascal's Triangle:

```c
#include <stdio.h>
long fun(int y)
{int z;
long result = 1;
for( z = 1 ; z <= y ; z++ )
result = result*z;
    return ( result );}
int main()
{int x, y, z;
printf("Input the number of rows in Pascal's triangle: ");
scanf("%d",&y);
for ( x = 0 ; x < y ; x++ )
{for ( z = 0 ; z <= ( y - x - 2 ) ; z++ )
printf(" ");
for( z = 0 ; z <= x ; z++ )
printf("%ld ",fun(x)/(fun(z)*fun(x-z)));
printf("\n");}
return 0;}
```

C program for linear search using function:

```c
#include <stdio.h>
long linear_search(long [], long, long);
int main()
{long array[100], search, c, n, position;
 printf("Input number of elements in array\n");
   scanf("%ld", &n);
   printf("Input %d numbers\n", n);
   for (c = 0; c < n; c++)
    scanf("%ld", &array[c]);
   printf("Input number to search\n");
   scanf("%ld",&search);
   position = linear_search(array, n, search);
   if (position == -1)
printf("%d is not present in array.\n", search);
   else
printf("%d is present at location %d.\n", search, position+1);
   return 0;}
long linear_search(long a[], long n, long find) {
   long c;
for (c = 0 ;c < n ; c++ ) {
if (a[c] == find)
return c;}
return -1;}
```

Function to check vowel:

```c
int check_vowel(char a)
{if (a >= 'A' && a <= 'Z')
    a = a + 'a' - 'A';
if (a == 'a' || a == 'e' || a == 'i' || a == 'o' || a == 'u')
       return 1;
    return 0;
}
```

C program for merge two sorted array using function:

```c
#include<stdio.h>
#include<conio.h>
void merge(int a[10],int b[10],int n,int m);
void main()
{int a[10],b[10],i,j,n,m;
 printf("Enter the limit of the first array");
 scanf("%d",&n);
 printf("Enter the first array");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 printf("Enter the limit of the second array");
 scanf("%d",&m);
 printf("Enter the second array");
 for(j=0;j<n;j++)
 scanf("%d",&b[j]);
 merge(a,b,n,m);
 getch();}
void merge(int a[],int b[],int n,int m)
 {int c[15],i,j;
   for(i=0;i<n;i++)
   {c[i]=a[i];}
   for(i=0;i<n;i++)
   {printf("%d",c[i]);}
   for(i=n,j=0;j<m,i<m+n;i++,j++)
   {c[i]=b[j];}
    printf("merge array");
    for(i=0;i<n+m;i++)
    printf("%d\t",c[i]);}
```

C Program to Find the Nth Fibonacci Number using function:

```c
#include <stdio.h>
int fibo(int);
int main()
{int num;
int result;
 printf("Enter the nth number in fibonacci series: ");
scanf("%d", &num);
if (num < 0)
{printf("Fibonacci of negative number is not possible.\n");}
    else
{result = fibo(num);
printf("The %d number in fibonacci series is %d\n", num, result);}
    return 0;}
int fibo(int num)
{if (num == 0)
    {return 0;}
    else if (num == 1)
    {return 1;}
    else
    {return(fibo(num - 1) + fibo(num - 2));}}
```