***Programming***

for

***problem Solving***

***in C***

Solutions to the questions in pps lab of jntuh syllabus(CS206ES 2018 -19)

(Common to CSE,EEE and IT)

By yours

***Sathish***

# PREFACE

:::::::::::::::::::::::::::::::::::::::::::::◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆::::::::::::::::::::::::::::::::::::::

This book have been chiefly designed to support students in understanding programming in a better way and definitely not as a source for them copy/rot without working on the programs .Most of the programs recorded in this book were really tried by me and my friends. I tried to record everything that is possible. While I have endeavoured to make the solutions mistake-free, some errors might have crept in. I would appreciate hearing from you. If you find any mistakes that have crept in please let me know. I will rectify those in the upcoming editions. Thank you....🙂

**Dedicated To**

**Everyone Reading this....**😄

Third Edition :-

After rectifying all those errors in the first and second editions and with added programs in the section - II, the third edition is ready to rock...

Thank you all for the support....

**Index**

**SL.No Program Title Pg. No**

1. Arithmetic, bitwise operators 04
2. Type conversion and casting 07
3. Smallest largest of three 08
4. Simple & Compound interest 08
5. Declare class awards. 09
6. Multiplication table. 09
7. Binary equivalent of decimal num 10
8. Physics problem s=ut+1/2at^2. 11
9. Calculations with switch. 12
10. Prime number check . 13
11. Sum of digits & palindrome check 14
12. Fibonacci sequence 15
13. Print prime numbers . 16
14. Roots of quadratic equation 17
15. 1-x/2+x^2/4-x^3/6 series 18
16. 1+×+×^2+×^3..... . 18
17. Min,Max and avg of array. 19
18. Mean variance deviation 20
19. Matrix addition. 21
20. Matrix Multiplication. 22
21. Transpose of a matrix 24
22. Factorial. 26
23. Gcd 26
24. x^n . 27
25. Pointer input 29
26. Pointer reverse output 29
27. Sum using pointer var. 30
28. File output 30
29. File copy with uppercase 31
30. Search ch in a file using cmdlin 32
31. bin file create and alter 33
32. Merge data into third file 34
33. Count ch, words, lines 35
34. Roman to decimal 36
35. Decimal to roman . 37
36. Sub string to main string. 39
37. Delete n characters at x 40
38. String Palindrome. 41
39. Position of ch. 42
40. Misc small, large, avg. 43
41. Number pyramids 44
42. Linear search function 48
43. Binary search function 49
44. Bubble sort 50
45. Selection sort . 51
46. Insertion sort. 52
47. Sorting array of names. 53

**1.Write a simple program that prints the results of all the operators available in C (including pre/ post increment , bitwise and/or/not , etc.). Read required operand values from standard input.**

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a+b;

printf("sum of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a-b;

printf("sub of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a\*b;

printf("mul of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a/b;

printf("div of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a%b;

printf("modulo div of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a&b;

printf("bitwise of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a|b;

printf("bitwise of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=a^b;

printf("bitwise of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,c;

printf("enter the values");

scanf("%d",&a);

c=a<<1;

printf("bitwise of %d is %d",a,c);

}

#include<stdio.h>

void main()

{

int a,c;

printf("enter the values");

scanf("%d",&a);

c=a>>1;

printf("bitwise of %d is %d",a,c);

}

#include<stdio.h>

void main()

{

int a,c;

printf("enter the values");

scanf("%d",&a);

c= ~a;

printf("bitwise of %d is %d",a,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=b+ ++a;

printf("sum of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=b+ a++;

printf("sum of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=b+ --a;

printf("sum of %d and %d is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter the values");

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

c=b+ a--;

printf("sum of %d and %d is %d",a,b,c);

}

**2. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values form standard input.**

#include<stdio.h>

void main()

{

int a,c;

float b;

printf("enter the values");

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

c=a+b;

printf("sum of %d and %f is %d",a,b,c);

}

#include<stdio.h>

void main()

{

int c;

float a,b;

printf("enter the values");

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

c=(int)a+(int)b;

printf("sum of %f and %f is %d",a,b,c);

}

**3. Write a c program for finding the max and min from given three numbers.**

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter three numbers");

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

if(a>b&&a>c)

{

printf("%d is the greatest number among three\n",a);

}

if(b>c&&b>a)

{

printf("%d is the greatest number among three\n",b);

}

if(c>b&&c>a)

{

printf("%d is the greatest number among three\n",c);

}

if(a<b&&a<c)

{

printf("%d is the smallest number among three\n",a);

}

if(b<c&&b<a)

{

printf("%d is the smallest number among three\n",b);

}

if(c<b&&c<a)

{

printf("%d is the smallest number among three\n",c);

}

}

**4. Write a c program for calculating simple and compound interest.**

#include<stdio.h>

#include<math.h>

void main()

{

float p,t,r,ci,si;

printf("enter amount time and rate");

scanf("%f %f %f",&p,&t,&r);

si=(p\*t\*r)/100;

ci=(p\*pow((1+r/100),t));

printf("Simple interest is %f",si);

printf("Compound interest is %f",ci);

}

**5. Write a C Program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.**

#include<stdio.h>

void main()

{

int c;

printf("enter percentage");

scanf("%d",&c);

if(c>=70)

{

printf("Distinction");

}

if(c<70&&c>=60)

{

printf("Second class");

}

if(c<60&&c>=40)

{

printf("Third class");

}

else if(c<40)

{

printf("fail");

}

}

**6. Write a program that prints a multiplication table for a given number up to a given number of rows.**

of rows in the table.#include<stdio.h>

void main()

{

int n,a,c,r;

printf("enter number and no.of rows to be printed");

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

for(a=0;a<=r;a++)

{

c=n\*a;

printf("%d x %d = %d\n",n,a,c);

}

}

**7.Write a program that shows the binary equivalent of a given positive number between 0 to 255.**

#include<stdio.h>

void main()

{

int num,r,bin=0,base=1,a;

printf("enter a integer");

scanf("%d",&num);

a=num;

if(num>255)

{

printf("Beyond range");

}

else

{

while(num>0)

{

r=num%2;

bin=bin+r\*base;

num=num/2;

base=base\*10;

}

printf("The binary equivalent of %d is %d",a,bin);

}

}

**8. A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula s = ut+(1/2)at^2 where u and a are the initial velocity in m/sec (= 0) and acceleration in m/sec^2 (= 9.8 m/s^2)).**

#include<stdio.h>

#include<math.h>

int main()

{

int i,s,u=0;

float a=9.8,t;

for(s=3,i=9;s<=30;s=s+3)

{

t=(-u+(sqrt(u\*u+2\*a\*s)))/a;

printf("Ball took %f secs to reach %d floor\n",t,i);

}

}

**9. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,\*, /, % and use Switch Statement)**

#include<stdio.h>

void main()

{

int a,b,c;

char op;

printf("enter operator and values for a and b");

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

switch(op)

{

case'+': c=a+b;

printf("sum of %d and %d is %d",a,b,c);

break;

case'-': c=a-b;

printf("sub of %d and %d is %d",a,b,c);

break;

case'\*': c=a\*b;

printf("product of %d and %d is %d",a,b,c);

break;

case'/': c=a/b;

printf("div of %d and %d is %d",a,b,c);

break;

case'%': c=a%b;

printf("rem of %d and %d is %d",a,b,c);

break;

default:printf("Invalid operator");

break;

}

}

**10. Write a program that finds if a given number is a prime number or composite number.**

#include<stdio.h>

void main()

{

int num,i,count=0;

printf("enter a integer");

scanf("%d",&num);

if(num<2)

{

printf("%d is neither prime nor composite",num);

}

else

{

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

{

if(num%i==0)

{

count++;

}

}

if(count==2)

{

printf("%d is a prime number",num);

}

else

{

printf("%d is a composite number",num);

}

}

}

**11. Write a C program to find the sum of individual digits of a positive integer and test the number for palindrome.**

#include<stdio.h>

void main()

{

int num,sum=0,rev=0,r,a;

printf("enter a number");

scanf("%d",&num);

a=num;

while(num>0)

{

r=num%10;

sum=sum+r;

rev=rev\*10+r;

num=num/10;

}

printf("sum of digits in %d is %d\n",a,sum);

if(a==rev)

{

printf("%d is a palindrome number",a);

}

else

{

printf("%d is not a palindrome number",a);

}

}

**12. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.**

#include<stdio.h>

void main()

{

int a,b,c,i,n;

printf("enter the range");

scanf("%d",&n);

printf("enter the first two numbers of the sequence");

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

printf("the sequence is ");

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

{

if(i<=1)

{

c=i;

}

else

{

c=a+b;

a=b;

b=c;

}

printf("%d,",c);

}

}

1**3. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.**

#include<stdio.h>

void main()

{

int a,i,n,count;

printf("enter the range");

scanf("%d",&n);

printf("prime numbers upto %d are ",n);

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

{

count=0;

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

{

if(a%i==0)

{

count++;

}

}

if(count==2)

{

printf("%d,",a);

}

}

}

**14. Write a C program to find the roots of a Quadratic equation.**

#include<stdio.h>

#include<math.h>

int main()

{

int a,b,c;

float d,x,y,imag;

printf("enter values for a,b,c in ax^2+bx+c");

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

d=b\*b-4\*a\*c;

if(d>0)

{

printf("the roots are %d/%d+%f/%d",-b,(2\*a),sqrt(d),2\*a);

printf("the roots are %d/%d-%f/%d",-b,(2\*a),sqrt(d),2\*a);

}

if(d==0)

{

printf("%d/%d is the root",-b,2\*a);

}

if(d<0)

{

x=y=-b/(2\*a);

imag=sqrt(-d)/(2\*a);

if(x>0)

{

printf("%f+%.2fi is the first root\n",x,imag);

printf("%f-%.2fi is the second root\n",y,imag);

}

else

{

printf("%d/%d+%.2fi is the first root\n",-b,2\*a,imag);

printf("%d/%d-%.2fi is the first root\n",-b,2\*a,imag);

}

}

}

**15. Write a C program to calculate the following, where x is a fractional value 1-x/2 +x^2/4-x^3/6.**

#include<stdio.h>

#include<math.h>

void main()

{

int n,m,q;

float s=0,p=0,x;

printf("enter value for x");

scanf("%f",&x);

printf("enter the range of series");

scanf("%d",&n);

for(m=1,q=2;m<=n;m=m+2,q=q+2)

{

s=-(pow((x),m))/(2\*m)+s;

p=(pow((x),q))/(2\*q)+p;

}

printf("The sum of the series is %f",s+p+1);

}

**16. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: 1+x+x^2+x^3+………….+x^n. For example: if n is 3 and x is 5, then the program computes 1+5+25+125.**

#include<stdio.h>

#include<math.h>

void main()

{

int n,m;

float s=0,x;

printf("enter value for x");

scanf("%f",&x);

printf("enter the range of series");

scanf("%d",&n);

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

{

s=(pow((x),m))+s;

}

printf("The sum of the series is %f",s+1);

}

**17. Write a C program to find the minimum, maximum and average in an array of integers.**

#include<stdio.h>

void main()

{

int a[100],i,n,small,large,avg=0;

printf("enter the range");

scanf("%d",&n);

printf("enter the elements");

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

{

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

}

small=large=a[0];

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

{

if(small>a[i])

{

small=a[i];

}

if(large<a[i])

{

large=a[i];

}

avg=avg+a[i];

}

avg=avg/n;

printf("%d is the minimum of array\n",small);

printf("%d is the maximum of array\n",large);

printf("%d is the average of array\n",avg);

}

**18.Write a functions to compute mean, variance, Standard Deviation, sorting of n elements in single dimension array.**

#include <stdio.h>

#include <math.h>

void main()

{

float x[100];

int i, n; float average, variance, std\_deviation, sum = 0, sum1 = 0;

printf("Enter the value of N \n");

scanf("%d", &n);

printf("Enter %d real numbers \n", n);

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

{

scanf("%f", &x[i]);

}

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

{

sum = sum + x[i];

}

average = sum / (float)n;

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

{

sum1 = sum1 + pow((x[i] - average), 2);

}

variance = sum1 / (float)n;

std\_deviation = sqrt(variance);

printf("Average of all elements = %.2f\n", average);

printf("variance of all elements = %.2f\n", variance);

printf("Standard deviation = %.2f\n", std\_deviation);

}

**19. Write a C program that uses functions to perform Addition of Two Matrices.**

#include<stdio.h>

void matadd(int a[10][10],int m,int n,int b[10][10],int p,int q)

{

int c[10][10],i,j;

if(n!=p)

{

printf("the matrices cannot be added");

}

else

{

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

{

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

{

c[i][j] = a[i][j] + b[i][j];

}

}

printf("the sum of the matrices is\n");

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

{

for(j=0;j<q;j++)

{

printf(" %d ",c[i][j]);

}

printf("\n");

}

}

}

void main()

{

int a[10][10],b[10][10],c[10][10],m,n,p,q,i,j,k,sum=0;

printf("enter the order of first matix\n");

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

printf("enter the elements of first matrix\n");

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

{

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

{

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

}

}

printf("enter the order of second matix\n");

scanf("%d %d",&p,&q);

printf("enter the elements of second matrix\n");

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

{

for(j=0;j<q;j++)

{

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

}

}

matadd(a,m,n,b,p,q);

}

**20. Write a C program that uses functions to perform multiplication of Two matrices.**

#include<stdio.h>

void matmul(int a[10][10],int m,int n,int b[10][10],int p,int q)

{

int c[10][10],i,j;

if(n!=p)

{

printf("the matrices cannot be multiplied");

}

else

{

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

{

for(j=0;j<q;j++)

{ sum =0;

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

{

Sum=sum+a[i][k]\*b[k][j];

} c[i][j]=sum;

}

}

printf("the product of the matrices is\n");

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

{

for(j=0;j<q;j++)

{

printf(" %d ",c[i][j]);

}

printf("\n");

}

}

}

void main()

{

int a[10][10],b[10][10],c[10][10],m,n,p,q,i,j,k,sum=0;

printf("enter the order of first matix\n");

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

printf("enter the elements of first matrix\n");

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

{

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

{

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

}

}

printf("enter the order of second matix\n");

scanf("%d %d",&p,&q);

printf("enter the elements of second matrix\n");

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

{

for(j=0;j<q;j++)

{

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

}

}

matmul(a,m,n,b,p,q);

}

**21. Write a C program that uses functions to perform transpose of a Matrix.**

#include<stdio.h>

void main()

{

int a[10][10],m,n,i,j;

printf("enter the order of matix\n");

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

printf("enter the elements of matrix\n");

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

{

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

{

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

}

}

printf("the original matrix is\n");

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

{

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

{

printf(" % d",a[i][j]);

}

printf("\n");

}

printf("the transpose of matrix is\n");

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

{

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

{

printf(" % d",a[j][i]);

}

printf("\n");

}

}

**22. Write C programs that use both recursive and non-recursive functions to find the factorial of a given integer.**

#include<stdio.h>

int rfact(int n)

{

int r;

if(n==1)

{

return 1;

}

r=n\*rfact(n-1);

return r;

}

int fact(int n)

{

int i,fct=1;

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

{

fct=fct\*i;

}

return fct;

}

void main()

{

int n,e,f;

char ch;

printf("eneter a number");

scanf("%d",&n);

printf("select recursive(a) or non recursive(b)");

scanf("%c",&ch);

scanf("%c",&ch);

switch(ch)

{

case'a':f=rfact(n);

break;

case'b':f=fact(n);

break;

default: print(“invalid input”);

break;

}

printf("The factorial of %d is %d",n,f);

}

**23. Write C programs that use both recursive and non-recursive functions to find the GCD (greatest common divisor) of two given integers.**

#include<stdio.h>

int rgcd (int a,int b)

{

int d;

if(b==0)

{

return a;

}

if(b>1)

{

d=a%b;

a=b;

b=d;

rgcd(a,b);

}

}

int gcd(int a,int b)

{

int d;

while(b>0)

{

d=a%b;

a=b;

b=d;

}

return a;

}

void main()

{

int a,b,c;

char ch;

printf("enter two numbers");

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

printf("select recursive(a) or non recursive(b)");

scanf("%c",&ch);

scanf("%c",&ch);

switch(ch)

{

case'a':c=rgcd(a,b);

break;

case'b':c=gcd(a,b);

break;

default:printf("invalid input");

break;

}

printf("gcd of the numbers is %d",c);

}

**24. Write C programs that uses both recursive and non-recursive functions to find x^n.**

#include<stdio.h>

int rxpown(int x,int n)

{

if(n==0)

{

return 1;

}

return x\*rxpown(x,n-1);

}

int xpown(int x,int n)

{

int i,res=1;

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

{

res=res\*x;

}

return res;

}

void main()

{

int x,n,res;

char ch;

printf("enter a number and power");

scanf("%d %d",&x,&n); printf("select recursive(a) or non recursive(b)");

scanf("%c",&ch);

scanf("%c",&ch);

switch(ch)

{

case'a':res=rxpown(x,n);

break;

case'b':res=xpown(x,n);

break;

default:printf("invalid input");

break;

}

printf("%d power %d is %d",x,n,res);

}

**25. Write a program for reading elements using pointer into array and display the values using array.**

#include<stdio.h>

void main()

{

int a[10],n,\*p,i;

printf("enter the range of array");

scanf("%d",&n);

p=a;

printf("enter the elements");

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

{

scanf("%d",&\*(p+i));

}

printf("the elements are ");

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

{

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

}

}

**26. Write a program for display values reverse order from array using pointer.**

#include<stdio.h>

void main()

{

int a[10],n,\*p,i;

printf("enter the range of array");

scanf("%d",&n);

p=a;

printf("enter the elements");

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

{

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

}

printf("the elements in reverse order are ");

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

{

printf("%d ",\*(p+i));

}

}

**27. Write a program through pointer variable to sum of n elements from array.**

#include<stdio.h>

void main()

{

int a[10],n,\*p,i,sum=0;

printf("enter the range of array");

scanf("%d",&n);

p=a;

printf("enter the elements");

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

{

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

sum=sum+\*(p+i);

}

printf("the sum of the elements is %d",sum);

}

**28. Write a C program to display the contents of a file to standard output device.**

#include<stdio.h>

void main()

{

char ch,file[20];

FILE \*ftr;

strcpy(file,argv[1]);

ftr=fopen("file","r");

if(ftr==NULL)

{

printf("file doesn't exist");

}

else

{

while((ch=fgetc(ftr))!=EOF)

{

printf("%c",ch);

}

}

fclose (ftr);

}

**29. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.**

#include<stdio.h>

void main()

{

char ch;

FILE \*ftr,\*fte;

fte=fopen("D:/pps.txt","w");

ftr=fopen("D:/prop.txt","r");

if(ftr==NULL)

{

printf("file doesn't exist");

}

else

{

while((ch=fgetc(ftr))!=EOF)

{

if((int)ch>96&&ch<123)

{

ch=ch-32;

}

fputc(ch,fte);

}

}

fclose (ftr);

fclose (fte);

}

**30. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.**

#include<stdio.h>

void main(int argc,char \*argv[])

{

char ch,\*s;

int count=0;

FILE \*ftr;

ftr=fopen(argv[1],"r");

if(ftr==NULL)

{

printf("file doesn't exist");

}

else

{

s=argv[2];

while((ch=fgetc(ftr))!=EOF)

{

if(ch==\*s)

{

count++;

}

}

printf("%c is repeated %d times",\*s,count);

}

fclose (ftr);

}

**31. Write a C program that does the following:**

**It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function) Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function) The program should then read all 10 values and print them back.**

(Valid only for two digit integer input)

#include<stdio.h>

void main(int argc,char\*argv[])

{

int i,a[10],x,n;

char ch;

FILE \*ftr;

ftr=fopen(argv[1],"wb");

for(i=0;i<10;i++)

{

a[i]=atoi(argv[i+2]);

fprintf(ftr,"%d",a[i]);

}

printf("enter a value and position to replace");

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

rewind(ftr);

fseek(ftr,2\*x,SEEK\_SET);

fprintf(ftr,"%d",n);

fclose(ftr);

ftr=fopen(argv[1],"rb");

while((ch=fgetc(ftr))!=EOF)

{

printf("%c ",ch);

}

fclose(ftr);

}

**32. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).**

#include<stdio.h>

void main()

{

char ch;

FILE \*ftr,\*fts,\*ftt;

ftr=fopen("D:/pps.txt","r");

fts=fopen("D:/prop.txt","r");

ftt=fopen("D:/whatever.txt","w");

if(ftr==NULL||fts==NULL)

{

printf("file doesn't exist");

}

else

{

while((ch=fgetc(ftr))!=EOF)

{

fputc(ch,ftt);

}

while((ch=fgetc(fts))!=EOF)

{

fputc(ch,ftt);

}

fclose(ftr);

fclose (fts);

fclose (ftt);

}

}

**33. Write a C program to count the lines, words and characters in a given text.**

#include<stdio.h>

void main()

{

int a=0,b=0,c=0;

char ch;

FILE \*ftr;

ftr=fopen("D:/pps.txt","r");

if(ftr==NULL)

{

printf("file doesn't exist");

}

else

{

while((ch=fgetc(ftr))!=EOF)

{

c++;

if(ch==32)

{

a++;

c--;

}

if(ch=='\n'||ch==10)

{

b++;

c--;

}

}

printf("file has %d words\n",a+b+1);

printf("file has %d lines\n",b+1);

printf("file has %d characters\n",c);

}

fclose (ftr);

}

**34. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.**

#include<stdio.h>

#include<string.h>

void main()

{

int i,j,len,a[10],dec=0;

char rom[10];

printf("enter a roman number");

scanf("%s",&rom);

len=strlen(rom);

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

{

switch(rom[i])

{

case 'I':a[i]=1;

break;

case 'V':a[i]=5;

break;

case 'X':a[i]=10;

break;

case 'L':a[i]=50;

break;

default : printf("error");

break; }

}

dec= a[len-1];

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

{

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

{

dec=dec-a[i-1];

}

if(a[i]<=a[i-1])

{

dec=dec+a[i-1];

}

}

printf("%d is the decimal equivalent",dec);

}

**35. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent.**

#include<stdio.h>

void main()

{

int a,b,n,i;

char ch='I',bh='V',ah='X',dh='L';

printf("enter a number");

scanf("%d",&n);

a=n/10;

b=n%10;

printf("The roman number is ");

if(a<4)

{

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

{

printf("%c",ah);

}

}

if(a==4)

{

printf("%c%c",ah,dh);

}

if(a==5)

{

printf("%c",dh);

}

if(b<4)

{

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

{

printf("%c",ch);

}

}

if(b==4)

{

printf("%c%c",ch,bh);

}

if(b==5)

{

printf("%c",bh);

}

if(b>5&&b<9)

{

printf("%c",bh);

for(i=0;i<b-5;i++)

{

printf("%c",ch);

}

}

if(b==9)

{

printf("%c%c",ch,ah);

}

}

**36. Write a C program that uses functions to insert a sub-string in to a given main string from a given position.**

#include<stdio.h>

#include<string.h>

void insubm(char s[100],char a[100],int n)

{

int i,j;

char c[100];

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

{

c[i]=s[i];

}

for(i=0;i<strlen(a);i++)

{

c[n+i]=a[i];

}

for(i=n+strlen(a),j=n;i<strlen(a)+strlen(s);i++,j++)

{

c[i]=s[j];

}

printf("%s is the modified string",c);

}

void main()

{

int n;

char s[100],a[100];

printf("enter main string and sub-string");

scanf("%s %s",&s,&a);

printf("enter the position to insert");

scanf("%d",&n);

insubm(s,a,n);

}

**37. Write a C program that uses functions to delete n Characters from a given position in a given string.**

#include<stdio.h>

#include<string.h>

void delnstr(char s[100],int x,int n)

{

int i,j;

for(i=x;i<strlen(s);i++)

{

s[i]=s[i+n];

}

printf("%s is the modified string",s);

}

void main()

{

int n,x;

char s[100];

printf("enter a string");

scanf("%s",&s);

printf("enter the position and no.of characters to delete");

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

delnstr(s,x,n);

}

**38. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)**

#include<stdio.h>

#include<string.h>

void main()

{

int i,count=0;

char s[100];

printf("enter a string");

scanf("%s",s);

for(i=0;i<strlen(s)/2;i++)

{

if(s[i]==s[strlen(s)-1-i])

{

count++;

}

}

if(count==strlen(s)/2)

{

printf("palindrome");

}

else

{

printf("Not a palindrome");

}

}

**39. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn‘t contain ch.**

#include<stdio.h>

#include<string.h>

void main()

{

int i,x=-1;

char s[100],ch;

printf("enter a string");

scanf("%s",&s);

printf("enter the character to search");

scanf("%c",&ch);

scanf("%c",&ch);

for(i=0;s[i]!='\0';i++)

{

if(s[i]==ch)

{

x=i;

printf("The character is in %d position\n",x);

}

}

if(x==-1)

{

printf("%d",x);

}

}

**40. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.**

#include<stdio.h>

void main()

{

int a[100],n,i,j,large,small,sum,mul,avg,sort;

char ch;

printf("enter the no.of elements\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

printf("enter your choice smallest(a) largest(b) sum(c) avg(d) sort(e)");

scanf("%c",&ch);

scanf("%c",&ch);

switch (ch)

{

case 'a': small=a[0];

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

{

if(small>a[i])

{

small=a[i];

}

}

printf("%d is the smallest number",small);

break;

case 'b': large=a[0];

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

{

if(large<a[i])

{

large=a[i];

}

}

printf("%d is the largest number",large);

break;

case 'c':sum=0;

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

{

sum=sum+a[i];

}

printf("sum of numbers is %d",sum);

break;

case 'd':avg=0;

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

{

avg=avg+a[i];

}

avg=avg/n;

printf("average of numbers is %d",avg);

break;

default: printf(“error”); break;

case 'e':

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

{

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

{

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

{

sort=a[i];

a[i]=a[j];

a[j]=sort;

}

}

}

printf("The ascending order is\n");

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

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

printf("\n");

printf("The descending order is\n");

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

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

break;

}

}

**41. Write a C program to construct a pyramid of numbers.**

#include<stdio.h>

void main()

{

int i,j,n;

printf("enter the height of pyramid");

scanf("%d",&n);

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

{

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

{

printf("%d",i);

}

printf("\n");

}

}

#include<stdio.h>

void main()

{

int i,j,n;

printf("enter the height of pyramid");

scanf("%d",&n);

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

{

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

{

printf("%d",j);

}

printf("\n");

}

}

#include<stdio.h>

void main()

{

int i,j,n,a=1;

printf("enter the height of pyramid");

scanf("%d",&n);

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

{

for(j=1;j<=i;j++,a++)

{

printf(" %d ",a);

}

printf("\n");

}

}

#include<stdio.h>

void main()

{

int i,j,n,a=1;

printf("enter the height of pyramid");

scanf("%d",&n);

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

{

for(j=1;j<=i;j++,a++)

{

printf("\*");

}

printf("\n");

}

}

#include<stdio.h>

void main()

{

int i,j,n,a=1;

printf("enter the height of pyramid");

scanf("%d",&n);

for(i=1;i<=n-n/2;i++)

{

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

{

printf("\*");

}

printf("\n");

}

for(i=n/2;i>=0;i--)

{

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

{

printf("\*");

}

printf("\n");

}

}

**42. Write a C program that uses non recursive function to search for a Key value in a given list of integers using linear search method.**

#include<stdio.h>

int lsearch(int a[100],int n,int s)

{

int i;

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

{

if(a[i]==s)

{

return i;

}

}

return 999;

}

void main()

{

int a[100],i,n,s,l;

printf("enter the range\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

printf("enter the number to search\n");

scanf("%d",&s);

l=lsearch(a,n,s);

if(l==999)

{

printf("element not found");

}

else

{

printf("element found at %d position",l);

}

}

**43. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.**

#include<stdio.h>

int bsearch(int a[100],int n,int s)

{

int m,p=0,e=n-1,l=999;

while(p<=e && l==999)

{

m=(p+e)/2;

if(a[m]==s)

{

l=m;

return l;

}

if(a[m]<s)

{

p=m+1;

}

if(a[m]>s)

{

e=m-1;

}

}

return l;

}

void main()

{

int a[100],i,n,s,l;

printf("enter the range\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

printf("enter the number to search\n");

scanf("%d",&s);

l=bsearch(a,n,s);

if(l==999)

{

printf("element not found");

}

else

{

printf("element found at %d position",l);

}

}

**44. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.**

#include<stdio.h>

void main()

{

int a[100],n,i,j,temp;

printf("enter the range\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

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

{

for(j=0;j<n-i-1;j++)

{

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

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

printf("sorted elements are\n");

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

{

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

}

}

**45. Write a C program that sorts the given array of integers using selection sort in descending order.**

#include<stdio.h>

void main()

{

int a[100],i,j,k,n,large,temp;

printf("enter the range\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

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

{

large=a[i];

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

{

if(large<=a[j])

{

large=a[j];

k=j;

}

}

if(a[i]!=large)

{

temp=a[i];

a[i]=a[k];

a[k]=temp;

}

}

printf("the sorted elements are\n");

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

{

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

}

}

**46. Write a C program that sorts the given array of integers using insertion sort in ascending order.**

#include<stdio.h>

void main()

{

int a[100],n,i,j,temp;

printf("enter the range\n");

scanf("%d",&n);

printf("enter the elements\n");

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

{

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

}

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

{

temp=a[i];

j=i-1;

while(j>=0&&a[j]>temp)

{

a[j+1]=a[j];

j--;

}

a[j+1]=temp;

}

printf("sorted elements are\n");

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

{

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

}

}

**47. Write a C program that sorts a given array of names.**

#include<stdio.h>

#include<string.h>

int main()

{

int i,j,k,n;

char name[20][10],small[10],temp[10];

printf("enter tha number of strings");

scanf("%d",&n);

printf("enter %d strings",n);

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

{

scanf("%s",name[i]);

}

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

{

strcpy(small,name[i]);

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

{

if(strcmp(small,name[j])>=0)

{

strcpy(small,name[j]);

k=j;

}

}

if(strcmp(small,name[i])!=0)

{

strcpy(temp,name[i]);

strcpy(name[i],name[k]);

strcpy(name[k],temp);

}

printf("after sorting tha strings are \n");

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

{

printf("%s\n",name[i]);

}

}

}

The Solutions for Questions 18 and 31 are the best that I can give and might be ambiguous and inappropriate for some readers.

Suggestions/advices are always welcome...😇

Thank you.....Nd All the Best.......👍

**Section :- II**

**This section contains all the other programs we tried/solved which were part of our learning but were not part of the syllabus we followed.**

**INDEX :-**

1. **Armstrong number 55**
2. **Sum of even and odd num. 56**
3. **Predefined string functions 57**