**1. Program to solve Arithmetic Operations**

**Algorithm:**

Step 1: Start

Step 2: Read values for a & b

Step 3: Calculate s = a + b

Step 4: Calculate sb = a - b

Step 5: Calculate ml = a \* b

Step 6: Calculate dv = a / b

Step 7: Calculate md = a % b

Step 8: Print s

Step 9: Print sb

Step 10: Print ml

Step 11: Print dv

Step 12: Print md

Step 13: Stop

**Program:**

#include<stdio.h>

#include<conio.h>

main()

{

int a, b, s, sb, ml, dv, md;

clrscr();

printf(“Enter the values for a & b:”);

scanf(“%d %d”, &a, &b);

s = a + b;

sb = a - b;

ml = a \* b;

dv = a / b;

md = a % b;

printf(“The Sum of %d & %d is: %d\n”, a, b, s);

printf(“The Difference of %d & %d is: %d\n”, a, b, sb);

printf(“The Product of %d & %d is: %d\n”, a, b, ml);

printf(“The Quotient of %d & %d is: %d\n”, a, b, dv);

printf(“The Remainder of %d & %d is: %d”, a, b, md);

getch();

}

**OUTPUT:**

Enter the values for a & b: 5

3

The Sum of 5 & 3 is: 8

The Difference of 5 & 3 is: 2

The Product of 5 & 3 is: 15

The Quotient of 5 & 3 is: 1

The Remainder of 5 & 3 is: 2

**2. Program to maintain student database**

**Algorithm:**

Step1: start

Step 2: read rno, nm, s1, s2, s3

Step 3: calculate tot = s1 + s2 + s3

Step 4: avg = tot / 3

Step 5: print rno

Step 6: print nm

Step 7: print s1, s2, s3

Step 8: print tot, avg

Step 9: check if s1 >= 35 and s2 >= 35 and s3 >= 35 then goto step10 otherwise step 15

Step 10: if avg >= 60 then goto step 11 otherwise goto step 12

Step 11: print ”first class” goto step 16

Step 12: check if avg >= 50 and avg < 60 then goto 13 else goto 14

Step 13: print “second class” goto step 16

Step 14: print “third class” goto step 16

Step 15: print “fail”

Step 16: stop

**Program:**

#include<stdio.h>

#include<conio.h>

main()

{

int rno, s1, s2, s3, tot;

float avg;

char nm[20];

printf(“Enter the Roll Number, Name, Subject1, Subject2 and Subject3”);

scanf(“%d %s %d %d %d”, &rno, nm, &s1, &s2, &s3);

tot = s1 + s2 + s3;

avg = (float)tot/3;

printf(“\n roll no:%d”, rno);

printf(“\n name:%s”, nm);

printf(“\n marks :%d \t %d \t %d”, s1, s2, s3);

printf(“\n total:%d \n average:%f”, tot, avg);

printf(“\n Grade: “);

if((s1 >= 35) && (s2 >= 35) && (s3 >= 35))

{

if(avg >= 60)

printf(“First Class”);

else

if((avg >= 50) && (avg < 60))

printf(“Second Class”);

else

printf(“Third Class”);

}

else

printf(“Fail”);

getch();

}

**Output:**

Enter the Roll Number, Name, Subject 1, Subject 2 and Subject 3:

roll no : 101

name : ram

marks : 45 50 48

total : 143

average : 44.3

Grade : Third Class

**3. PROGRAM TO FIND WHETHER A GIVEN NUMBER IS AMSTRONG OR NOT**

**ALGORITHM:**

Step 1: Start

Step 2: Read 'n' value

Step 3: t = n, rev = 0

Step 4: Loop

Step 5: While(n>0) goto Step 6 else Step 10

Step 6: dig = n%10

Step 7: rev = rev + (dig\*dig\*dig)

Step 8: n = n/10

Step 9: goto Step 5

Step 10: if(t == rev) goto Step 11 else goto Step 12

Step 11: Print "The given number is Amstrong" then goto Step13

Step 12: Print "The given number is not Amstrong"

Step 13: Stop

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main( )

{

int n, t, rev, dig;

clrscr( );

printf("Enter the number:");

scanf("%d", &n);

t = n;

rev = 0;

while(n>0)

{

dig = n % 10;

rev = rev + (dig \* dig \* dig);

n = n / 10;

}

if(t = = rev)

printf("The given number is amstrong");

else

printf("The given number is not Armstrong");

getch( );

}

**OUTPUT:**

Enter the number: 153

The given number is Armstrong

**4. PROGRAM TO FIND THE RANGE OF EVEN NUMBERS**

**ALGORITHM:**

Step 1: Start

Step 2: Read 'n' value

Step 3: Initialize i with 1

Step 4: Repeat until i<n

4.1: if ((i % 2) = = 0) is true

4.2: then print i

4.3: increase i by 1 until i <= n

Step 5: Stop

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main( )

{

int n, i;

clrscr( );

printf("Enter the range:");

scanf("%d", &n);

i=1;

while(i < = n)

{

if((i % 2) = = 0)

printf("%d\t", i);

i ++;

}

getch( );

}

**OUTPUT:**

Enter the range: 5

1. 4 6 8 10

**5. PROGRAM TO FIND THE RANGE OF PRIME NUMBERS**

**ALGORITHEM:**

Step 1: Start

Step 2: Read value for r

Step 3: Initialize i with 1

Step 4: Repeat until i <= r

4.1: Initialize j = 1, C = 0

4.2: Repeat until j <= i

4.2.1: If i % j = = 0 is true then

4.2.2: c = c + 1

4.2.3: j = j + 1 until j <= i

4.3: if c = = 2 is true then

4.4: Print i

4.5: Increase i by 1 unit i <= r

Step 5: Stop

**Program:**

#include<stdio.h>

#include<conio.h>

main( )

{

Int i, j, r, c;

clrscr( );

printf("Enter the range:");

scanf("%d", &r);

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

{

c=0;

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

{

if( i % j = = 0)

c = c + 1;

}

if(c= =2)

printf("%d\t", i);

}

getch( );

}

OUTPUT:

Enter the range: 5

1. 3 5 7

**6. PROGRAM TO PRINT FEBBINACCI SERIES**

**ALGORITHEM:**

Step 1: Start

Step 2: Read 'n' value

Step 3: pre=0, pra =1

Step 4: Print pre, pra

Step 5: i=1

Step 6: if(i<=n-2) then goto step 7 else goto step 12

Step 7: pri = pre + pra

Step 8: Print pri

Step 9: pre = pra

Step 10: pra = pri

Step 11: pre = pre + 1 then goto step 6

Step 12: Stop

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main( )

{

int f1, f2, f, i, n;

clrscr( );

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

scanf("%d", &n);

f1 = 0;

f2 =1;

printf("%3d%3d", f1, f2);

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

{

f = f1 + f2;

printf("%3d", f);

f1 = f2;

f2 = f;

}

getch( );

}

**OUTPUT:**

Enter the value of n: 5

1. 1 1 2 3 5 8

**7. PROGRAM TO SORT THE GIVEN NUMBERS USING BUBBLE SORT METHOD**

**ALGORITHEM:**

Step 1: Start

Step 2: Read n value

Step 3: for(i=0;i<n; i++)

(i) Read a[i]

Step 4: for(i=0;i<n-1;i++)

Step 4.1: for(j=i+1;j<n; j++)

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

i) t = a[i]

ii) a[ i ] = a[j]

iii) a[ j ] = t

Step 5: for(i=0; i<n; i++)

i) Print a[i]

Step 6: Stop

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main( )

{

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

clrscr( );

printf("Enter the range:");

scanf("%d",&n);

printf("Enter %d numbers:",n);

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

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

printf("The values before sorting are:\n");

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

{

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

printf("\n");

}

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

{

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

{

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

{

t = a[i];

a[i] = a[j];

a[j] = t;

}

}

}

printf("The values After sorting are:\n");

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

{

printf("\n");

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

}

getch( );

}

**OUTPUT:**

Enter the range: 5

Enter 5 numbers: 25 10 15 5 30

The values before sorting are: 25 10 15 5 30

The values after sorting are: 5 10 15 25 30

**8. PROGRAM TO PERFORM MATRIX MULTIPLICATION AND PRINT RESULTANT MATRIX**

**ALGORITHEM:**

Step 1: Start

Step 2: Read r1, c1

Step 3: for(i=0;i<r1;i++)

3.1 for(j=0;j<c1;j++)

i) Read a[i][j]

Step 4: Read r2, c2

Step 5: for(i=0;i<r2;i++)

5.1 for(j=0;j<c2;j++)

i) Read b[i][j]

Step 6: if(c1==r2) then goto step 7 else goto step 10

Step 7: for(i=0;i<r1;i++)

(a) c[i][j]=0

(b) for(k=0;k<r2;k++)

i) c[i][j] = c[i][j] + a[i][k] \* b[k][j];

Step 8: for(i=0;i<r1;i++)

8.1 for(j=0;j<c2;j++)

( a ) Print c[i][j]

( b ) Print("\n")

Step 9: Goto Step 11

Step 10: Print Multiplication is not possible

Step 11: Stop

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main( )

{

int a[10][10], b[10][10], c[10][10], r1, c1, r2, c2, i, j, k;

clrscr( );

printf("Enter the first row and column size: ");

scanf("%d%d",&r1,&c1);

printf("Enter the values: ");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

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

}

}

printf("\nEnter the second row and column size: ");

scanf("%d%d",&r2, &c2);

printf("Enter the values: ");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

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

}

}

if(c1==r2)

{

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

c[i][j]=0;

for(k=0;k<r2;k++)

{

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

}

}

}

printf("\n The Resultant matrix A\*B = :\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

printf("%3d\t",c[i][j]);

}

printf("\n");

}

}

else

printf("Multiplication is not possible");

getch( );

}

OUTPUT:

Enter the first row and column size: 2 2

Enter the values: 1 2 1 2

Enter the second row and column size: 2 2

Enter the values: 2 1 2 1

The Resultant matrix A\*B = 2 2

1. 2

**9. PROGRAM TO SEARCH AN ELEMENT IN THE ARRAY**

**ALGORITHM:**

Step 1: start

Step 2: declare a[100] as an array of integers

Step 3: read n value

Step 4: flag = 0

Step 5: for(i=0;i<n; i++)

5.1: read number for a[i]

Step 6: read ‘num’

Step 7: for(i=0;i<n;i++)

7.1: if(a[i]==num)then goto step 7.2 else goto step7

7.2: flag=1

Step 8: if(flag==1goto step 9 else goto step 10

Step 9: print number found got o step 11

Step 10: print number not found

Step 11: stop

**Program:**

#include<stdio.h>

#include<stdio.h>

main()

{

int a[100],n, num, i, flag = 0;

clrscr();

printf(“enter n value:”);

scanf(“%d”, &n);

printf(“enter %d numbers”, n);

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

scanf(“%d”, &a[i]);

printf(“enter the search number”);

scanf(“%d”, &num);

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

{

if(a[i]==num)

{

flag=1;

}

}

if(flag==1)

printf(“number found and it’s position is %d”,i);

else

printf(“number not found”);

getch();

}

Output:

enter n value:5

enter 5 numbers: 5 20 15 10 30

enter your number: 15

number found and its position is 3

**10. PROGRAM TO FIND WHETHER THE STRING IS PALINDROME OR NOT**

**Algorithm:**

Step 1: start

Step 2: read string 1

Step 3: copy string 1 to string 2

Step 4: reverse string 2

Step 5: if(string 1==string 2)goto 6 else goto 7

Step 6: print “string is palindrome’goto step 8

Step 7: print”string is not palindrome”

Step 8: stop

**Program:**

#include<stdio.h>

#include<string.h>

main()

{

char s1[20],s2[20];

int n;

clrscr();

printf(“enter your string”);

gets(s1);

strcpy(s2,s1);

strrev(s2);

n = strcmp(s1,s2);

if(n == 0)

printf(“\n string is palindrome”);

else

printf(“\n string is not palindrome”);

getch();

}

Output:

enter your string : Malayalam

string is palindrome

**11. PROGRAM TO SWAP TWO NUMBERS USING CALL BY REFERENCE**

**Algorithm:**

Step 1: start

Step 2: read a, b

Step 3: print a, b

Step 4: call swap(&a, &b)

Step 5: print a,b

Step 6: stop

Title: swap(\*a,\*b)

Step 1: t=\*p;

Step 2: \*p=\*q

Step 3: \*q=t

Step 4: goto step 4 in main()

Program:

main()

{

int a, b;

clrscr();

printf(“enter 2 no’s”);

scanf(“%d %d”, &a, &b);

printf(“\n a:%d \n b:%d”, a, b);

swap(&a, &b);

printf(“\n a:%d \n b:%d” ,a, b);

getch();

}

swap (int \*p, int \*q)

{

int t;

t = \*p;

\*p = \*q;

\*q = t;

}

Output:

Enter 2 no’s: 2 3

a: 2 b: 3

a: 3 b: 2

**12. PROGRAM TO COPY A FILE**

**Algorithm:**

Step 1: start

Step 2: open file 1 in read mode and assign adderess to f1

Step 3: open file 2 in read mode and assign adderess to f2

Step 4: while(ch=fget((\*f1))!=eof)

4.1 fputc(ch,f2)

Step 5: fclose(f1)

Step 6: fclose(f2)

Step 7: print data coppied

Step 8: stop

**Program:**

#include<stdio.h>

#include<conio.h>

main()

{

char ch;

FILE \*f1, \*f2;

clrscr();

if(((f1 = fopen(“file 1.txt”, ”r”)) == NULL) || ((f2 = fopen(“file2.txt”, ”w”)) == NULL

{

printf(“file can’t be open”);

exit(0);

}

else

{

while((ch = fget c(f1))! = eof)

{

fput c(ch,f2);

}

printf(“\n file copied”);

}

fclose(f1);

fclose(f2);

getch();

}

Output:

file coppied

**Aim: Program to find the roots of Quadratic Equation**

**Algorithm:**

#include<stdio.h>

#include<conio.h>

main()

{

float a,b,c,d,x1,x2,rp,ip;

clrscr();

printf(“enter a,b,c values”);

scanf(“%f%f%f,&a,&b&c);

If(a==0)

{

(x1=(-c/b);

printf(“Roots:%\f”,x1);

}

else

{

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

If(d>=o)

{

x1=(-b+sqrt(d))/(2\*a);

x2=(-b-sqrt(d))/(2\*a);

printf(“\n roots are real:\n x1:%\f \n x2:%\f”,x1x2);

}

else

{

xp=-b/(2\*a);

ip=root-d/(2\*a);

printf(“\n imaginary roots”);

printf(“\n x1=%\f+i%\f”,rp,ip);

printf(“\n x2=%\f-i%\f”,rp,ip);

}  
}  
getch();

}

**Output:**

Enter a,b,c values

1

2

3

imaginary roots

X1=1.000000+i-16287.000000

X2=1.000000-i-16287.000000