**Dr. D.Y. Patil Pratishthan's**

**Institute for Advanced Computing and Software Development**

**(IACSD), Akurdi, Pune**

**Assignments and their Solutions (C and C++)**

**Name of Candidate:- Manjiri Raut**

**Roll no :- 1110**

**----------- Assignments on Arrays in C Programming -----------**

**1. Find all odd and even no.'s in runtime array?**

#include<stdio.h>

int main(

{

    int i, num;

    printf("Enter Array Size:");

    scanf("%d", &num);

    int \*array = malloc( num \* sizeof(int) );

    printf("\nEnter Array Elements:\n");

    for(i=0; i<num; i++)

    {

        scanf("%d", &array[i]);

    }

    printf ("\nEven elements are: ");

    for(i=0; i < num; i++)

    {

        if(array[i] % 2 == 0)

        {

            printf("%d\t",array[i]);

        }

    }

    printf ("\n\nOdd elements are: ");

    for(i=0; i < num; i++)

    {

        if(array[i] % 2 != 0)

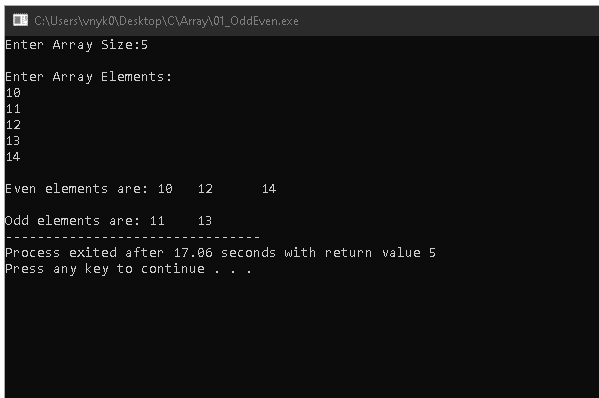
        {

            printf("%d\t",array[i]);

        }

    }

}



**2. Find all prime no.'s in runtime array?**

#include<stdio.h>

int main()

{

    int n,i,j;

    printf("\nEnter array size: ");

    scanf("%d",&n);

int \*a = malloc(n \* sizeof(int));

printf("\nEnter array elements:\n");

    for(i=0; i<n; i++)

    {

         scanf("\t%d", &a[i]);

    }

    printf("\nYour Array: ");

    for(i=0; i<n; i++)

    {

        printf("\t%d", a[i]);

    }

    printf("\n\nPrime numbers in the Array: ");

    for(i=0; i<n; i++)

    {

     int count=0;

     for(j=2; j<a[i]; j++)

     {

     if(a[i]%j==0)

     {

     count=1;

     break;

}

}

if(count==0)

{

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

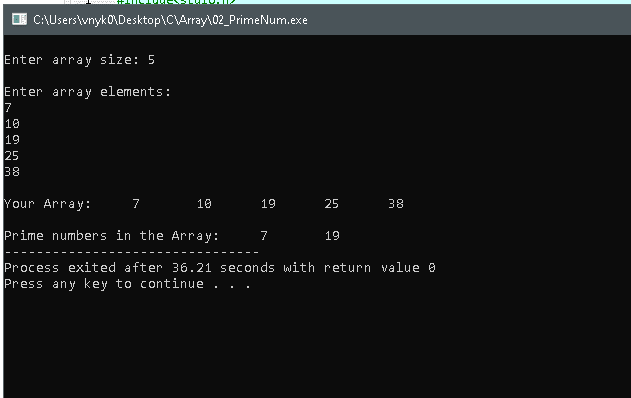
}

}

    getch();

    return 0;

}



**3. Write a program to create an array of integers and perform following operations on that array like finding the sum, average, maximum and minimum number in that array. Accept the numbers of the array from user.**

#include<stdio.h>

int main()

{

int num, i, sum, avg;

sum=0;

printf("\nEnter array size:\t");

scanf("%d", &num);

int \*arr = malloc( num \* sizeof(int) );

printf("\nEnter array elements:\n");

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

{

scanf("\n%d",&arr[i]);

}

printf("\nYour array: ");

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

{

printf("\t%d",arr[i]);

}

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

{

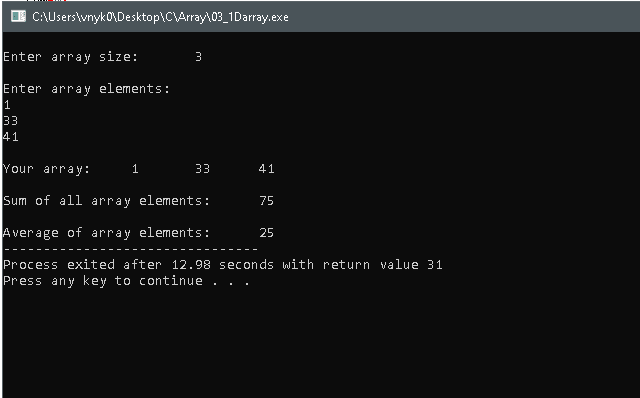
sum = sum + arr[i];

}

printf("\n\nSum of all array elements:\t%d",sum);

avg=sum/num;

printf("\n\nAverage of array elements:\t%d",avg); }



**4. Perform all the above operations for 2D arrays.**

#include<stdio.h>

int sum(int a[3][3]);

void average();

int main()

{

int arr[3][3],j,i, add;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

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

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

}

}

printf("\n2D array..\n");

    for(i=0;i<3;i++)

    {

        printf("\n");

        for (j=0;j<3;j++)

        {

            printf("%d\t",arr[i][j]);

        }

    }

    add=sum(arr);

printf("\n\nSum of all array elements:\t%d",add);

average(add);

getch();

}

int sum(int a[3][3])

{

int i,j, sum=0;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

sum = sum + a[i][j];

}

}

return sum;

}

void average(int a)

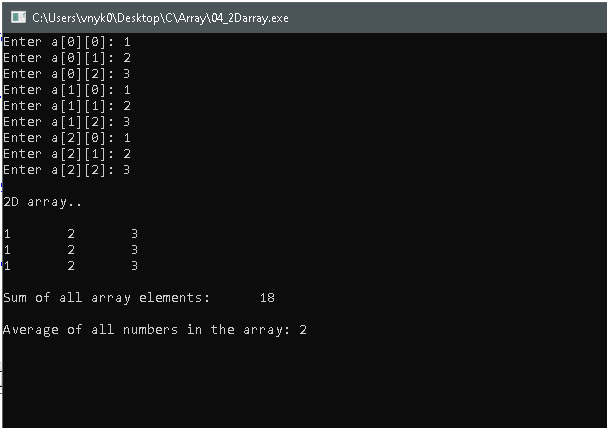
{

int avg;

avg= a/9;

printf("\n\nAverage of all numbers in the array: %d", avg);

}



**5. Enter data for two matrices. Multiply them to store result in third matrix & display result.**

#include <stdio.h>

int main()

{

  int m, n, p, q, c, d, k, sum = 0;

  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter number of rows and columns of first matrix : ");

  scanf("%d%d", &m, &n);

  printf("Enter elements of first matrix: ");

  for (c = 0; c < m; c++)

    for (d = 0; d < n; d++)

      scanf("%d", &first[c][d]);

  printf("Enter number of rows and columns of second matrix :");

  scanf("%d%d", &p, &q);

if (n != p)

  printf("The multiplication isn't possible.\n");

  else

  {

    printf("Enter elements of second matrix :");

    for (c = 0; c < p; c++)

      for (d = 0; d < q; d++)

        scanf("%d", &second[c][d]);

    for (c = 0; c < m; c++) {

      for (d = 0; d < q; d++) {

        for (k = 0; k < p; k++) {

          sum = sum + first[c][k]\*second[k][d];

        }

        multiply[c][d] = sum;

        sum = 0;

      }

    }

 printf("Product of the matrices:\n");

    for (c = 0; c < m; c++) {

      for (d = 0; d < q; d++)

        printf("%d\t", multiply[c][d]);

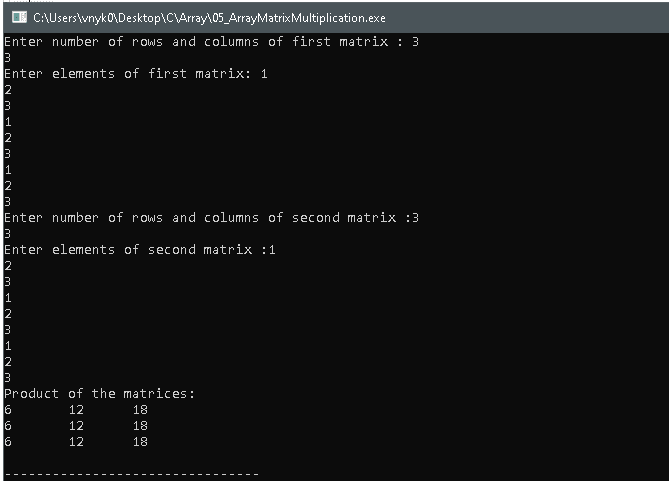
      printf("\n");

    }

  }

  return 0;

}



**6. Enter data for one matrix. Find its Transpose & display result.**

#include <stdio.h>

int main()

{

int row,col,i,j,a[row][col],transpose[col][row],k,l;

printf("Enter row size of 2D array to perform transpose operation: ");

scanf("%d",&row);

printf("Enter column size of 2D array to perform transpose operation: ");

scanf("%d",&col);

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

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

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

printf("\nTranspose Matrix: \n");

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

{

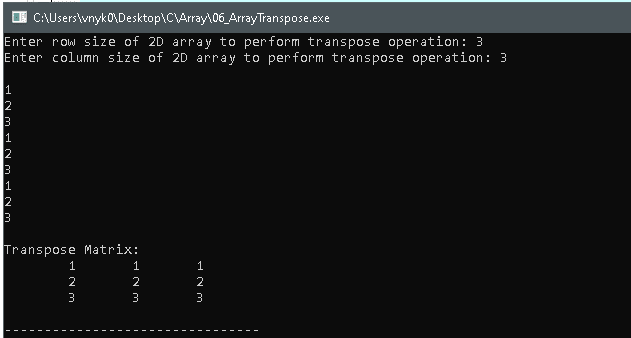
for(l=0;l<col;l++)

printf("\t%d",transpose[k][l]=a[l][k]);

printf("\n");

}

}



**----------- Assignments on Decision and Loops -----------**

**1. Write a C program to input basic salary of an employee and calculate its Gross salary according to following: Basic Salary <= 10000 : HRA = 20%, DA = 80% Basic Salary <= 20000 : HRA = 25%, DA = 90% Basic Salary > 20000 : HRA = 30%, DA = 95%**

#include<stdio.h>

int main()

{

float gross,basic,hra,da;

printf("Please enter basic salary to calculate gross salary :");

scanf("%f",&basic);

if(basic<=10000)

{

hra=basic\*0.2;

da=basic\*0.8;

}

else if(basic<=20000)

{

hra=basic\*0.9;

da=basic\*0.25;

}

else

{

da=basic\*0.95;

hra=basic\*0.3;

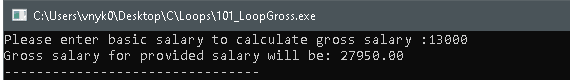
}

gross=basic+hra+da;

printf("Gross salary for provided salary will be: %.2f",gross);

return 0;

}



**2. Write a C program to input angles of a triangle and check whether triangle is valid or not.**

#include<stdio.h>

void checkTriangle(int a[3]);

int main() //main Function

{

int arr[3],i;

printf("\nEnter three angles of the triangle:\n");

for(i=0; i<3; i++)

{

scanf("\n%d", &arr[i]); //Accepting three angles of the triangle from use

}

checkTriangle(arr); //Passing an array to the function

return 0;

}

void checkTriangle(int a[3]) //Function to check if triangle is valid or not

{

int i, sumOfAngles=0;

for(i=0;i<3;i++)

{

sumOfAngles = sumOfAngles + a[i];

}

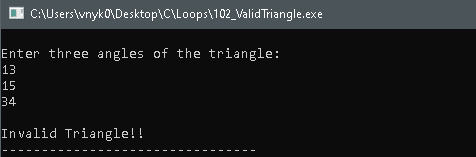
if(sumOfAngles==180)

printf("\nThis triangle is valid!!");

else

printf("\nInvalid Triangle!!");

}



**3. Accept a number and display it multiplication table.**

#include<stdio.h>

int main()

{

int num,i;

printf("\nEnter a number:\t");

scanf("%d", &num);

for (i = 1; i <= 10; i++)

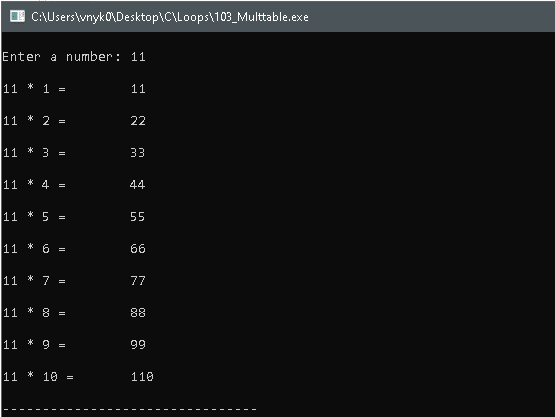
{

        printf("\n%d \* %d =\t%d \n", num, i, num \* i);

    }

return 0;

}



**4. Accept a number and display its sum of digits.**

#include<stdio.h>

int main()

{

int num,sum;

sum =0;

printf("Enter the number:\t");

scanf("%d",&num);

while(num!=0)

{

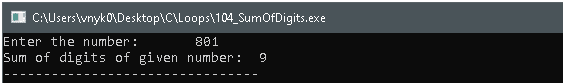
sum= sum + (num % 10);

num=num/10;

}

printf("Sum of digits of given number:\t%d",sum );

}



**5. Accept a number and display whether its an Armstrong number.**

#include<stdio.h>

void armStrong(int);

int main()

{

int num;

printf("\nEnter a three digit number to check if its Armstring: ");

scanf("\t%d", &num);

armStrong(num);

return 0;

}

void armStrong(int n)

{

int remainder, c=0, inputNum;

inputNum = n;

while(n != 0)

{

remainder = n %10;

c += remainder \* remainder \* remainder;

n=n/10;

}

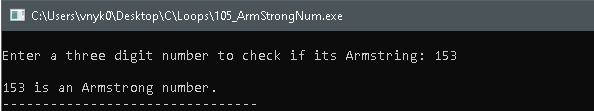
if(c == inputNum)

printf("\n%d is an Armstrong number.", inputNum );

else

printf("\n%d is not a Armstrong number.", inputNum );

}



**6. Write a C program to check whether a character is uppercase or lowercase alphabet.**

#include<stdio.h>

void checkChar(char);

void main() //main Function

{

char ch;

printf("\nEnter an alphabet:\t");

scanf("%c",&ch);

checkChar(ch); //passing character to function

}

void checkChar(char c) //Function to check upper and lower case alphabet

{

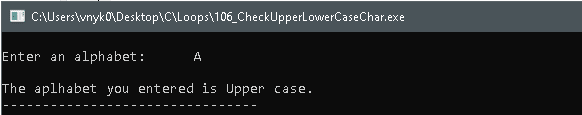
if(c>='a' && c<='z')

printf("\nThe aplhabet you entered is Lower case.");

else

printf("\nThe aplhabet you entered is Upper case.");

}



**7. Write a C program to input week number and print week day.**

#include <stdio.h>

void checkWeekday(int);

int main()

{

  int weekday;

  printf(" Please Enter the Day Number 1 to 7:  ");

  scanf("%d", &weekday);

  checkWeekday(weekday);

  return 0;

}

void checkWeekday(int day)

{

if (day == 1)

  {

   printf("\n Today is Monday.");

  }

  else if ( day == 2 )

  {

   printf("\n Today is Tuesday.");

  }

  else if ( day == 3 )

  {

   printf("\n Today is Wednesday.");

  }

  else if ( day == 4 )

  {

   printf("\n Today is Thursday.");

  }

  else if ( day == 5 )

  {

   printf("\n Today is Friday.");

  }

  else if ( day == 6 )

  {

   printf("\n Today is Saturday.");

  }

  else if ( day == 7 )

  {

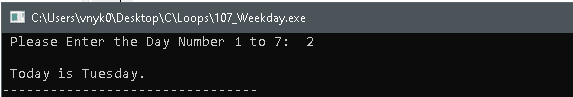
   printf("\n Today is Sunday.");

  }

  else

    printf("\n Please enter Valid Number between 1 to 7.");

}



**8. Write a C program to input month number and print number of days in that month.**

#include<stdio.h>

int main()

{

int num;

char \*(a[12])={"Jan","Feb","March","Apr","May","Jun","Jul","Aug","Sept","Oct","Nov","Dec"};

printf("Please enter the month number: ");

scanf("%d",&num);

if (num>0 || num<13)

{

if(num==2)

printf("Feb has either 28 or 29 days");

else if(num%2==0)

printf("%s has days 30",a[num]);

else

printf("%s has days 31",a[num-1]);

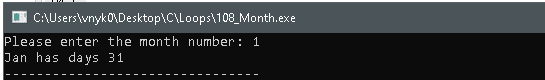
}

else

printf("Month number should be can be either or between 1 and 12 ");

return 0;

}



**9. Write a C program to count total number of notes in given amount.**

#include<stdio.h>

void perform();

void operation(int);

int main()

{

perform();

return 0;

}

void perform()

{

int amount;

printf("\nPlease enter the amount : ");

scanf("%d",&amount);

operation(amount);

}

void operation(int amount)

{

int temp,a[]={2000,500,200,100,50,20,10,5,2,1},i;

temp=amount;

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

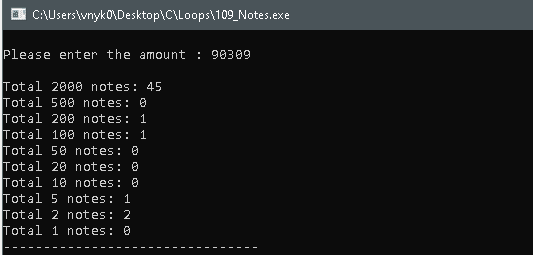
{

printf("\nTotal %d notes: %d",a[i],temp/a[i]);

temp%=a[i];

}

}



**10. Write a C program to invert the case of alphabet.**

#include<stdio.h>

#include<stdlib.h>

char get();

void change(char);

int main()

{

change(get());

return 0;

}

char get()

{

char c;

printf("\nPlease enter a character: ");

scanf("%c",&c);

return c;

}

void change(char c)

{

if(c>='A' && c<='Z')

c = 'a' + (c - 'A');

else if(c >= 'a' && c <='z')

c = 'A' + (c - 'a');

else

{

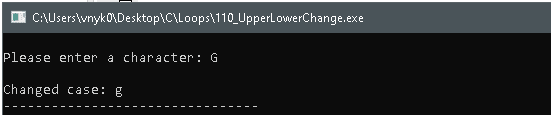
printf("\nInvalid input");

exit(0);

}

printf("\nChanged case: %c ",c);

}



**11. Write a C program to print all natural numbers from 1 to n. - using while loop**

#include<stdio.h>

int get();

void find(int);

int main()

{

find(get());

return 0;

}

int get()

{

int n;

printf("\nPlease enter number to find natural numbers from 1 to n: ");

scanf("%d",&n);

return n;

}

void find(int n)

{

int i=1;

while(i<=n)

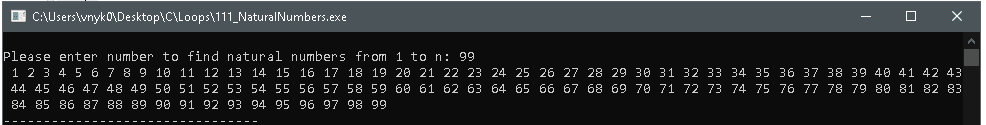
{

printf(" %d",i);

i++;

}

}



**12. Write a C program to print all natural numbers in reverse (from n to 1). - using while loop**

#include<stdio.h>

int get();

void find(int);

int main()

{

find(get());

return 0;

}

int get()

{

int n;

printf("\nPlease enter number to find natural numbers from n to 1: ");

scanf("%d",&n);

return n;

}

void find(int n)

{

while(1<=n)

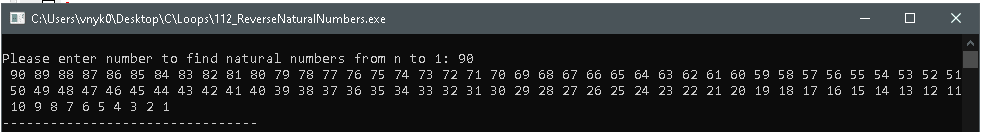
{

printf(" %d",n);

n--;

}

}



**13. Write a C program to find sum of all even and odd numbers between 1 to n.**

#include<stdio.h>

int get();

void find(int);

int main()

{

find(get());

return 0;

}

int get()

{

int n;

printf("\nPlease enter number to find addition of even and odd numbers: ");

scanf("%d",&n);

return n;

}

void find(int n)

{

int counter=1,evenSum=0,oddSum=0;

while(counter<=n)

{

if(counter%2==0)

evenSum+=counter;

else

oddSum+=counter;

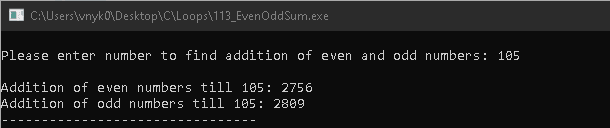
counter++;

}

printf("\nAddition of even numbers till %d: %d",n,evenSum);

printf("\nAddition of odd numbers till %d: %d",n,oddSum);

}



**14. Write a C program to count number of digits in a number.**

#include<stdio.h>

int get();

void find(int);

int main()

{

find(get());

return 0;

}

int get()

{

int n;

printf("\nEnter a number: ");

scanf("%d",&n);

return n;

}

void find(int n)

{

int counter=0;

while(n!=0)

{

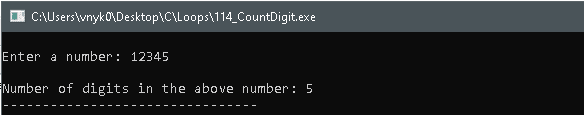
n/=10;

counter++;

}

printf("\nNumber of digits in the above number: %d",counter);

}



**15. Write a C program to find first and last digit of a number.**

#include<stdio.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("Enter a number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int LastDigit = 0, FirstDigit = 0;

LastDigit = n % 10;

while(n != 0)

{

FirstDigit = n;

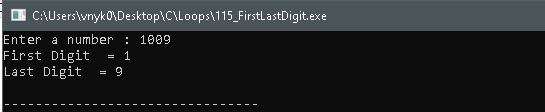
n/=10;

}

printf("First Digit  = %d\n",FirstDigit);

printf("Last Digit  = %d\n",LastDigit);

}



**16. Write a C program to enter a number and print its reverse.**

#include<stdio.h>

int get();

int reverse(int);

int main()

{

reverse(get());

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

printf("\Reverse : %d",reverse(n));

}

int rev=0;

int reverse(int n)

{

int rem;

if(n)

{

rem=n%10;

rev=rev\*10+rem;

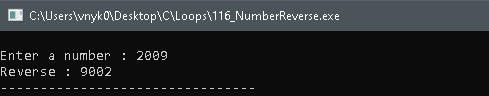
}

else

      return rev;

return rev;

}



**17. Write a C program to check whether a number is palindrome or not.**

#include<stdio.h>

void get();

int check(int);

int main()

{

get();

return 0;

}

void get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

int result=check(n);

if(result==0)

printf("\nThe number is palindrome");

else

printf("\nThe number is not palindrome");

}

int check(int n)

{

int original=n,lastDigit=0,rev=0,result=1;

while(n!=0)

{

lastDigit=n%10;

rev=rev\*10+lastDigit;

n/=10;

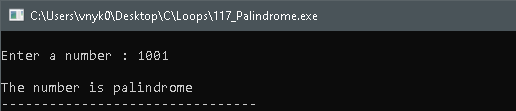
}

if(rev==original)

result=0;

return result;

}



**18. Write a C program to find power of a number using for loop.**

#include<stdio.h>

int get();

int find(int,int);

int main()

{

get();

return 0;

}

int get()

{

int n,power;

printf("Enter a number : ");

scanf("%d", &n);

printf("Enter power : ");

scanf("%d", &power);

printf("result: %d",find(n,power));

}

int find(int n,int power)

{

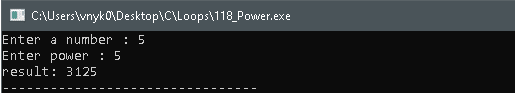
int i,result=1;

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

result=result\*n;

return result;

}



**19. Write a C program to find all factors of a number.**

#include<stdio.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nenter number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int i;

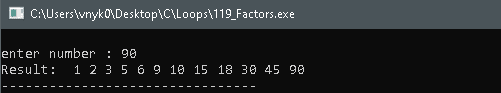
printf("Result: ");

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

if(n%i==0)

printf(" %d",i);

}



**20. Write a C program to calculate factorial of a number.**

#include<stdio.h>

int get();

int fact(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

printf("Factorial %d is %d",n,fact(n));

}

int fact(int n)

{

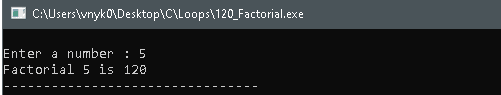
if(n>=1)

return n\*fact(n-1);

else

return 1;

}



**21. Write a C program to find HCF (GCD) of two numbers.**

#include<stdio.h>

int get();

int HCF(int,int);

int main()

{

get();

return 0;

}

int get()

{

int n,m;

printf("\nEnter first number : ");

scanf("%d", &n);

printf("\nEnter second number : ");

scanf("%d", &m);

printf("\nHCF : %d",n,m,HCF(n,m));

}

int HCF(int n,int m)

{

int gcd=0,i,min=0;

min=n<m?n:m;

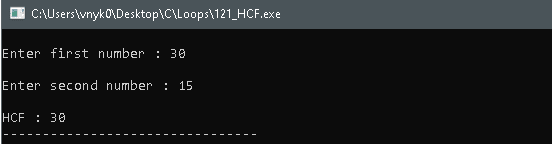
for(i=1;i<=(min/2);++i)

if(n%i==0 && m%i==0)

gcd=i;

return gcd;

}



**22. Write a C program to find LCM of two numbers.**

#include<stdio.h>

int get();

int LCM(int,int);

int main()

{

get();

return 0;

}

int get()

{

int n,m;

printf("\nEnter first number : ");

scanf("%d", &n);

printf("\nEnter second number : ");

scanf("%d", &m);

LCM(n,m);

}

int LCM(int n,int m)

{

int min=n<m?n:m,lcm,i,gcd;

for(i=1;i<=(min/2);++i)

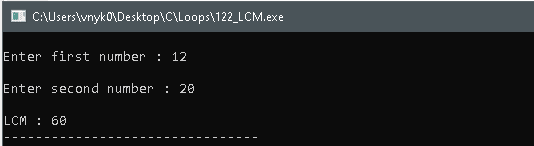
if(n%i==0 && m%i==0)

gcd=i;

lcm=(n\*m)/gcd;

printf("\nLCM : %d",lcm);

}



**23. Write a C program to check whether a number is Prime number or not.**

#include<stdio.h>

#include<stdlib.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int i=2,counter=0;

while(i<=n/2)

{

if(n%i==0)

{

counter=1;

break;

}

++i;

}

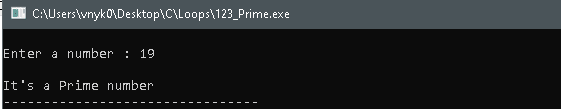
if(counter==0)

printf("\nIt's a Prime number");

else

printf("\nIt's not a Prime number");

}



**24. Write a C program to print all Prime numbers between 1 to n.**

#include<stdio.h>

#include<stdlib.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int j=2;

while(j<=n)

{

int i=2,counter=0;

while(i<=j/2)

{

if(j%i==0)

{

counter=1;

break;

}

++i;

}

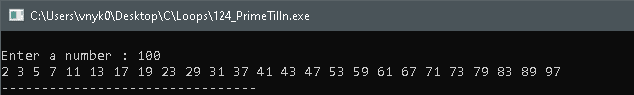
if(counter==0)

printf("%d ",j);

++j;

}

}



**25. Write a C program to check whether a number is Perfect number or not.**

***Perfect number* is a positive integer which is equal to the sum of its proper positive divisors. For example: 6 is the first perfect number Proper divisors of 6 are 1, 2, 3 Sum of its proper divisors = 1 + 2 + 3 = 6. Hence 6 is a perfect number.**

#include<stdio.h>

#include<stdlib.h>

int get();

int find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

find(n);

}

int find(int n)

{

int i,sum=0;

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

{

if(n%i==0)

sum+=i;

}

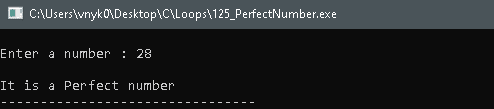
if(sum==n)

printf("\nIt is a Perfect number");

else

printf("\nIt is not a Perfect number");

}



**26. Write a C program to check whether a number is Strong number or not.**

***Strong number* is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since, 1! + 4! + 5! = 145**

#include<stdio.h>

#include<stdlib.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int original=n,i,fact=1,lastdigit=0,sum=0;

while(n!=0)

{

fact=1;

lastdigit=n%10;

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

fact\*=i;

sum+=fact;

n/=10;

}

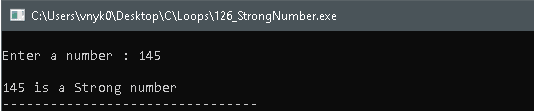
if(original==sum)

printf("\n%d is a Strong number",original);

else

printf("\n%d is not a Strong number",original);

}



**27. Write a C program to print Fibonacci series up to n terms.**

#include<stdio.h>

int get();

void find(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a number : ");

scanf("%d", &n);

find(n);

}

void find(int n)

{

int first=0,second=1,next=first+second,i;

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

    printf("%d",first);

while(first<n)

{

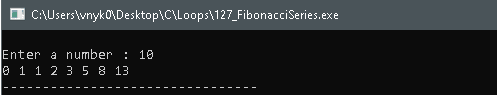
first=second;

second=next;

next=first+second;

printf(" %d",first);

}}



**28. Write a C program to print all alphabets from a to z**

**Write a function to generate the following pyramid of numbers**

**0**

**101**

**21012**

**3210123**

**432101234**

**54321012345**

**432101234**

**3210123**

**21012**

**101**

**0**

|  |  |
| --- | --- |
|  | |
| #include<stdio.h> | |
|  | void print(); | |
|  | int main() | |
|  | { | |
|  | print(); | |
|  | return 0; | |
|  | } | |
|  | void print() | |
|  | { | |
|  | int i=0; | |
|  | printf("\nA-Z letters:\n"); | |
|  | for(i=0;i<26;++i){ | |
|  | printf(" %c",'A'+i); | |
|  | printf("\n\na-z letters:\n");  } | |
|  | for(i=0;i<26;++i) | |
|  | printf(" %c",'a'+i); | |
|  | | **}**  **Output:** |

#include <stdio.h>

void get();

void print(int);

int main()

{

get();

return 0;

}

void get(){

int n;

printf("\nEnter Total Number of Rows :");

scanf("%d",&n);

print(n);

}

void print(int n)

{

int i,j,k,l;

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

{

for(j = n\*2; j >= 0; j--)

if(i == j)

printf("%d",0);

else if(i > j)

printf("%d",i - j);

else if( i < j && (j - i <= i))

printf("%d",j - i);

else

printf(" ");

printf("\n");

}

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

for(j = n \* 2; j>=0; j--)

if(i == j)

printf("%d",0);

else if(i > j)

printf("%d",i -j);

else if(i < j && (j-i <= i))

printf("%d", j-i);

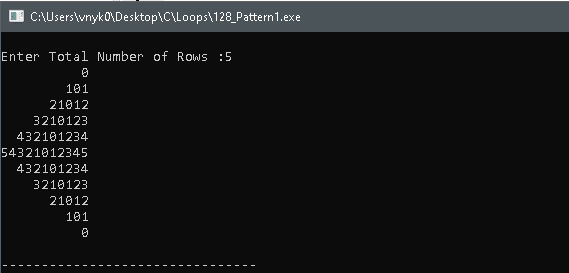
else

printf(" ");

printf("\n");

}

}



**29. Get this output using simple loops. Here no of iterations are important.**

**zyxwvwxyz**

**zyxwxyz**

**zyxyz**

**zyz**

**z**

#include<stdio.h>

int get();

void print(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nEnter a Number : ");

scanf("%d", &n);

print(n);

}

void print(int n)

{

int i,j,whitespace;

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

{

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

printf(" ");

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

printf("%c",123-j);

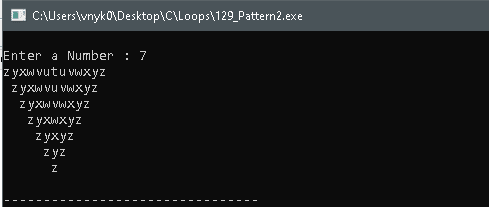
for(j=j-2;j>=1;j--)

printf("%c",123-j);

printf("\n");

}

}



**------- Assignments on Strings in C Programming -------**

**1: Write a program to sort a number of strings using bubble sort. Input is a number of strings and the output is the sorted list based on the length of strings.**

**For e.g.: If input is jyoti, sareeka, anisha, sangita, savita, suja**

**The output is suja, jyoti, anisha, savita, sareeka, sangita**

 #include <stdio.h>

 #include <string.h>

    void main()

    {

        char name[10][8], tname[10][8], temp[8];

        int i, j, n;

        printf("Enter size of array: \n");

        scanf("%d", &n);

        printf("\nEnter names: ");

        for (i = 0; i < n; i++)

        {

            scanf("%s", name[i]);

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

        }

        for (i = 0; i < n - 1 ; i++)

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

                if (strcmp(name[i], name[j]) > 0)

                {

                    strcpy(temp, name[i]);

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

                    strcpy(name[j], temp);

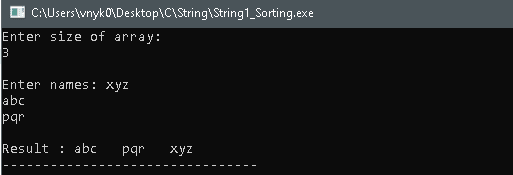
                }

printf("\nResult : ");

        for (i = 0; i < n; i++)

            printf("%s   ", name[i]);

    }



**2: Define a structure to represent time in hours (0-23), minutes**

**(0-59), and seconds (0-59), and then write a function that accepts an argument of type time represented by this structure and updates it by one second & 30 seconds.**

#include <stdio.h>

typedef struct Time{

int hours;

int minutes;

int seconds;

}Time;

void updateMin(int,Time\*);

void updateSec(int,Time\*);

void setTime(Time\*);

void setHr(Time\*\*);

void setMin(Time\*\*);

void setSeconds(Time\*\*);

int main()

{

int hr = 0,mm = 0, ss = 0;

printf(" Hours: ");

scanf("%d",&hr);

printf(" Minutes: ");

scanf("%d",&mm);

printf(" Seconds: ");

scanf("%d",&ss);

Time t1  = {hr,mm,ss};

setTime(&t1);

printf("Your Time:\t%d %d  %d \n",t1.hours,t1.minutes,t1.seconds );

updateMin(1,&t1);

updateSec(30,&t1);

printf("Updated Time:\t%d %d  %d \n",t1.hours,t1.minutes,t1.seconds );

}

void setTime(Time \*temp)

{

setHr(&temp);

setMin(&temp);

setSeconds(&temp);

}

void setHr(Time \*\*temp)

{

int hr = (\*temp) -> hours;

if(hr >=0 && hr <24)

(\*temp) -> hours = hr;

else

(\*temp) -> hours = 0;

}

void setMin(Time \*\*temp)

{

int mm = (\*temp) -> minutes;

int r1 = 0;

if(mm >=0 && mm <60)

(\*temp)->minutes = mm;

else

{

r1 = mm/60;

if((\*temp)->hours < 24)

{

(\*temp)->hours += r1;

(\*temp)->minutes = mm % 60;

}

if((\*temp)->hours > 23)

{

(\*temp)->hours = 0;

}

}

}

void setSeconds(Time \*\*temp)

{

int ss = (\*temp) ->seconds;

int r2 = 0;

if(ss >=0 && ss < 60)

(\*temp)->seconds = ss;

else

{

r2 = ss /60;

if((\*temp)->minutes <= 59)

{

(\*temp)->minutes += r2;

(\*temp)->seconds = ss % 60;

}

if((\*temp)->minutes > 59)

{

(\*temp)->hours+=((\*temp)->minutes/60);

(\*temp)->minutes = 0;

}

if((\*temp)->hours > 23)

(\*temp)->hours = 0;

}

}

void updateMin(int mm, Time \*temp)

{

int iTemp = (temp) -> minutes;

(temp) -> minutes = iTemp+ mm;

if((temp)->minutes > 59)

setHr(&temp);

}

void updateSec(int ss, Time \*temp)

{

int iTemp = temp -> seconds;

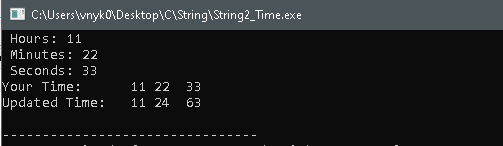
temp -> seconds = iTemp + ss;

int iMin = (temp -> seconds) / 60;

if(temp -> seconds > 59)

updateMin(iMin,temp);

}



**3: Using pointers write your own functions for the following:**

**a. String comparison b. String concatenate**

**c. String copy**

**d. String length.**

**Note: Do not include <string.h> in your program**

#include<stdio.h>

#include<stdlib.h>

int Compare(char\*, char\*);

void Concat(char\*, char\*);

void Copy(char\*, char\*);

int Length(char\*);

int main()

{

int result = 0;

int choice = 0;

int done = 1;

char string1[20] = {'\0'};

char string2[20] = {'\0'};

while(done){

printf("\nChoose: \n");

printf("1.  comparison\n");

printf("2.  concatenate\n");

printf("3.  copy\n");

printf("4.  length\n");

printf("0. exit\n");

scanf("%d",&choice);

switch(choice){

case 1:

printf("Enter: ");

scanf(" %[^'\n']s",string1);

printf("Enter: ");

scanf(" %[^'\n']s",string2);

result = Compare(string1,string2);

if(result == 0)

printf("Equal\n");

else

printf("Not equal\n");

break;

case 2:

printf("Enter: ");

scanf(" %[^'\n']s",string1);

printf("Enter: ");

scanf(" %[^'\n']s",string2);

Concat(string1,string2);

printf(" Merge:  %s\n",string1);

break;

case 3:

printf("Enter: ");

scanf(" %[^'\n']s",string1);

Copy(string2,string1);

printf("Copy %s: \n",string2);

break;

case 4:

printf("Enter: ");

scanf(" %[^'\n']s",string1);

printf("Length %d\n",Length(string1));

    break;

case 0: done = 0;

break;

}

}

}

int Compare(char \*str1, char \*str2)

{

int flag = 0;

if(str1 == NULL || str2 == NULL)

return -1;

while(( \*str1 != '\0' )&& (\*str2 != '\0'))

{

if(\*str1 !=  \*str2)

{

flag = 1;

break;

}

str1++;

str2++;

}

if(flag == 1)

return 1;

else

return 0;

}

void Concat(char \*dest, char \*src)

{

if(dest == NULL || src == NULL)

return;

while(\*dest != '\0')

dest++;

while(\*src != '\0')

{

\*dest = \*src;

dest++;

src++;

}

\*dest = '\0';

}

void Copy(char \*dest, char \*src)

{

if(dest == NULL || src == NULL)

return;

while(\*src != '\0')

{

\*dest = \*src;

dest++;

src++;

}

\*src = '\0';

}

int Length(char \*str)

{

int count = 0;

if(str == NULL)

return -1;

while(\*str != '\0')

{

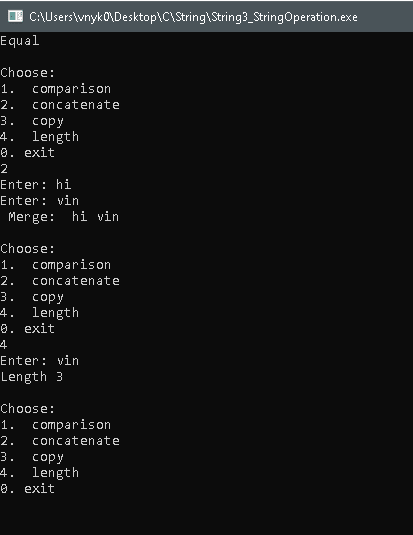
str++;

count++;

}

return count;

}



**4.Write a program that will read each line in a file and store it in another file with the sequence reversed ,that is,the first line in the file one should the last line in the file two and so on.**

#include <stdio.h>

int main(){

int n, reverseNumber = 0, rem,Originalnumber=0;

printf("Enter a number to get reverse number ");

scanf("%d", &n);

Originalnumber=n;

for(;n != 0;){

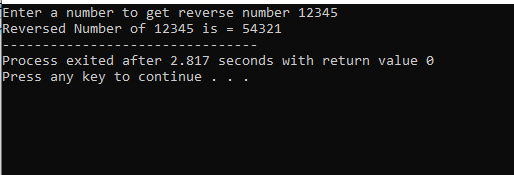
rem = n%10;

reverseNumber = reverseNumber\*10 + rem;

n /= 10;}

printf("Reversed Number of %d is = %d",Originalnumber,reverseNumber);

;}



**5: Write a program to cyclically permute a string one character at a time.**

**E.g.: If space is the input the output should produce**

**space paces acesp cespa espac**

#include<stdio.h>

#include <string.h>

void PermuteStr(char[]);

int main()

{

char arr[20] = {'\0'};

printf("Enter a word: ");

scanf("%[^'\n']s",arr);

PermuteStr(arr);

return 0;

}

void PermuteStr(char str[])

{

int len=strlen(str),i;

char t[len];

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

{

int j=i;

int k=0;

while(str[j]!='\0')

{

t[k]=str[j];

k++;

j++;

}

j = 0;

while(j<i)

{

t[k]=str[j];

j++;

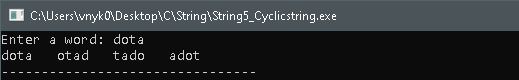
k++;

}

printf("%s   ",t);

}

}



**6: Write functions for the following base conversion operations: a. Octal to Hexadecimal. b. Hexadecimal to Octal.**

**Take care to validate digits/characters while accepting the input.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void OcttoHex();

void HextoOct();

int main()

{

    int ch=1;

    printf("\nPlease choose 1: \n1: Octal to Hex\n2: Hex to Octal\n0:Exit\n-> ");

    scanf("%d",&ch);

    switch(ch)

    {

     case 1:

     OcttoHex();

     break;

     case 2:

     HextoOct();

     break;

     case 0:

ch=0;

break;

     default:

     printf("try again");

    }

    return 0;

}

void OcttoHex()

{

int OctNumber, Reverse=0, LastDigit, counter=0, hex=0, mul=1, i=0, k=0;

    char BinaryNumber[40] = "", HexadecimalNumber[40];

printf("Enter  Number: ");

    scanf("%d", &OctNumber);

    while(OctNumber!=0)

    {

        LastDigit = OctNumber%10;

        if(LastDigit>7)

        {

            counter++;

            break;

        }

        Reverse = (Reverse\*10) + LastDigit;

        OctNumber = OctNumber/10;

    }

    if(counter==0)

    {

        OctNumber = Reverse;

        while(OctNumber!=0)

        {

            LastDigit = OctNumber%10;

            switch(LastDigit)

            {

                case 0: strcat(BinaryNumber, "000");

                    break;

                case 1: strcat(BinaryNumber, "001");

                    break;

                case 2: strcat(BinaryNumber, "010");

                    break;

                case 3: strcat(BinaryNumber, "011");

                    break;

                case 4: strcat(BinaryNumber, "100");

                    break;

                case 5: strcat(BinaryNumber, "101");

                    break;

                case 6: strcat(BinaryNumber, "110");

                    break;

                case 7: strcat(BinaryNumber, "111");

                    break;

            }

            OctNumber = OctNumber/10;

        }

        while(BinaryNumber[k]!='\0')

            k++;

        counter=1;

        k--;

        while(k>=0)

        {

            if(BinaryNumber[k]=='0')

                LastDigit = 0;

            else

                LastDigit = 1;

            hex = hex + (LastDigit\*mul);

            if(counter%4==0)

            {

                if(hex<10)

                    HexadecimalNumber[i] = hex+48;

                else

                    HexadecimalNumber[i] = hex+55;

                mul = 1;

                hex = 0;

                counter = 1;

                i++;

            }

            else

            {

                mul = mul\*2;

                counter++;

            }

            k--;

        }

        if(counter!=1)

            HexadecimalNumber[i] = hex+48;

        if(counter==1)

            i--;

        printf("Result = ");

        counter = 0;

        for(i=i; i>=0; i--)

        {

            if(HexadecimalNumber[i]=='0' && counter==0)

            {

                counter++;

                continue;

            }

            else

                printf("%c", HexadecimalNumber[i]);

        }

        printf("\n");

    }

}

void HextoOct()

{

int i=0, chk=0, len, rem, BinDigit, OctDigit;

    long long binaryInt=0,BinNumber, temp=1, OctNumber;

    char HexArray[11], BinArray[40]="";

    printf("Enter  Number: ");

scanf("%s",HexArray);

    while(HexArray[i])

    {

        switch(HexArray[i])

        {

            case '0':

                strcat(BinArray, "0000");

                break;

            case '1':

                strcat(BinArray, "0001");

                break;

            case '2':

                strcat(BinArray, "0010");

                break;

            case '3':

                strcat(BinArray, "0011");

                break;

            case '4':

                strcat(BinArray, "0100");

                break;

            case '5':

                strcat(BinArray, "0101");

                break;

            case '6':

                strcat(BinArray, "0110");

                break;

            case '7':

                strcat(BinArray, "0111");

                break;

            case '8':

                strcat(BinArray, "1000");

                break;

            case '9':

                strcat(BinArray, "1001");

                break;

            case 'A':

                strcat(BinArray, "1010");

                break;

            case 'a':

                strcat(BinArray, "1010");

                break;

            case 'B':

                strcat(BinArray, "1011");

                break;

            case 'b':

                strcat(BinArray, "1011");

                break;

            case 'C':

                strcat(BinArray, "1100");

                break;

            case 'c':

                strcat(BinArray, "1100");

                break;

            case 'D':

                strcat(BinArray, "1101");

                break;

            case 'd':

                strcat(BinArray, "1101");

                break;

            case 'E':

                strcat(BinArray, "1110");

                break;

            case 'e':

                strcat(BinArray, "1110");

                break;

            case 'F':

                strcat(BinArray, "1111");

                break;

            case 'f':

                strcat(BinArray, "1111");

                break;

            default:

                chk = 1;

                break;

        }

        i++;

    }

    if(chk==0)

    {

        len = strlen(BinArray);

        while(len!=0)

        {

            if(BinArray[len-1]=='0')

                BinDigit=0;

            else

                BinDigit=1;

            binaryInt = binaryInt + (BinDigit\*temp);

            temp = temp\*10;

            len--;

        }

        BinNumber = binaryInt;

        OctNumber = 0;

        temp = 1;

        while(BinNumber>0)

        {

            rem = BinNumber%1000;

            switch(rem)

            {

                case 0:

                    OctDigit = 0;

                    break;

                case 1:

                    OctDigit = 1;

                    break;

                case 10:

                    OctDigit = 2;

                    break;

                case 11:

                    OctDigit = 3;

                    break;

                case 100:

                    OctDigit = 4;

                    break;

                case 101:

                    OctDigit = 5;

                    break;

                case 110:

                    OctDigit = 6;

                    break;

                case 111:

                    OctDigit = 7;

                    break;

            }

            OctNumber = (OctDigit\*temp) + OctNumber;

            BinNumber = BinNumber/1000;

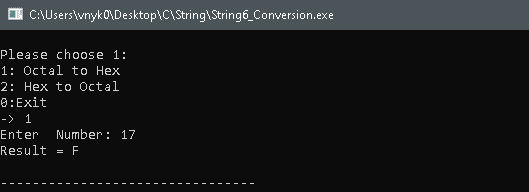
            temp = temp\*10;

        }

        printf("Result = %lld", OctNumber);

    }

}



**7: A program is to be written to implement the tower of Hanoi Problem.**

#include <stdio.h>

void tower(char,char,char,int);

int main()

{

int n;

printf("\nEnter number: ");

scanf("%d",&n);

tower('A','B','C',n);

return 0;

}

void tower(char from,char to,char other,int n)

{

if(n==1)

printf("\n\t %c to %c",from,other);

if(n>1)

{

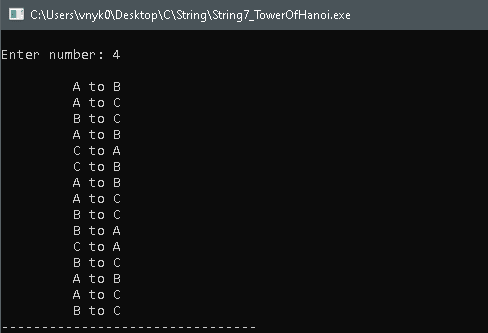
tower(from,other,to,n-1);

tower(from,to,other,1);

tower(to,from,other,n-1);

}

}



**--------------C++ Assignments on Classes and Objects--------------**

**1. Create a class Person with data members as name, age, city. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.**

 #include<string.h>

 #include<iostream>

using namespace std;

 class person

 {

  int age;

string name, city;

public:

person()

  {

  age=20;

  name="renuka default";

  city="Pune City default";

}

person(int age,string name,string city)

  {

  this->age=age;

  this->name=name;

  this->city=city;

}

void setAge(int age)

  {

  this->age=age;

  }

  void setName(string name)

  {

  this->name=name;

  }

  void setCity(string city)

  {

  this->city=city;

  cout<<city;

  }

  int getAge()

  {

  return age;

}

string getName()

{

return name;

}

string getCity()

{

return city;

}

//method to display values belonging to certain object

  void display()

  {

cout<<"name: "<<name<<"\nage:  "<<age<<"\nlocation: "<<city<<endl;

}

 };

 int main()

 {

  person p1;

  p1.display();

  p1.setName("renu");

  p1.getName();

  p1.setCity("Pune City");

  p1.getCity();

  p1.setAge(70);

  p1.getAge();

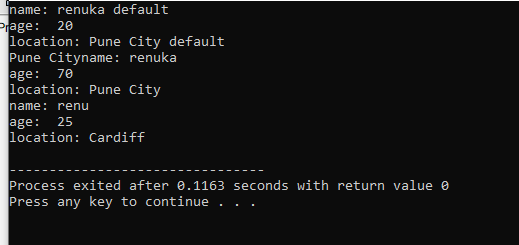
  p1.display();

person p2(25,"Vin","Cardiff");

p2.display();

return 0;

}



**2. Create a class Date with data members as dd, mm, yy. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.**

#include<iostream>\*/

using namespace std;

class date

{

   public:

        int dd, mm, yy;

        date()

        {

            this->dd=30;

            this->mm=04;

            this->yy=2021;

        }

        date (int d, int m, int y)

        {

            this->dd=d;

            this->mm=m;

            this->yy=y;

        }

       void setdd(int d)

        {

            this->dd=d;

        }

        void setmm(int m)

        {

            this->mm=m;

        }

        void setyy(int y)

        {

            this->yy=y;

        }

        int getdd()

        {

            this->dd;

        }

        int getmm()

        {

            this->mm;

        }

        int getyy()

        {

            this->yy;

        }

       void display()

        {

            cout<<"\nDate: "<<this->dd;

            cout<<"\nMonth: "<<this->mm;

            cout<<"\nYear: "<<this->yy;

        }

};

int main()

{

    date d1;

    d1.display();

    cout<<"\n";

    date d2(01, 05, 2021);

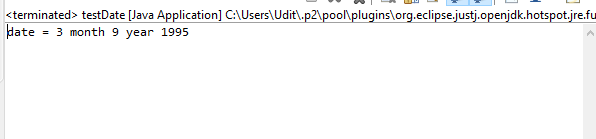
    d2.display();

    cout<<"\n";

    return 0;

}

**Output:**



**3. Create a class Book with data members as bname,id,author,price. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.**

#include<iostream>

#include<stdio.h>

using namespace std;

class date

{

int d,m,y;

public:

date()

{

this-> d=0;

this-> m=0;

t

}

date(int d,int m,int y)

{

this->d=d;

this->m=m;

this->y=y;

}

void setd(int d)

{

this-> d=d;

}

void setm(int m)

{

this-> m=m;

}

void sety(int y)

{

this-> y=y;

}

int getd()

{

return d;

}

int getm()

{

return m;

}

int gety()

{

return y;

}

void display()

{

cout<<d<<"/"<< m<<"/"<< y<<endl;

}

};

int main()

{

date d1;

d1.display();

d1.setd(16);

d1.getd();

d1.setm(9);

d1.getm();

d1.sety(1998);

d1.gety();

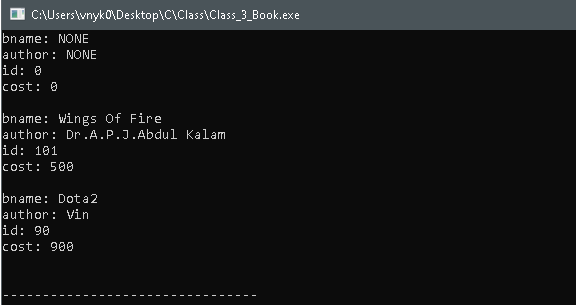
d1.display();

date d2(9,1,1996);

d2.display();

return 0;

}



**4. Create a class Point with data members as x,y. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.**

 #include<iostream>

 using namespace std;

 class point

 {

  int x,y;

public:

point()

{

this->x=0;

this->y=0;

}

point(int a,int b)

{

this-> x=a;

this-> y=b;

}

void setx(int a)

{

this->x=a;

}

void sety(int b)

{

this->y=b;

}

int getx()

{

return this-> x;

}

int gety()

{

return this-> y;

}

void display()

{

cout<<x<<"."<<y<<endl<<endl;

}

 };

 int main()

 {

  point p;

  p.display();

p.setx(99);

p.getx();

p.sety(1);

p.gety();

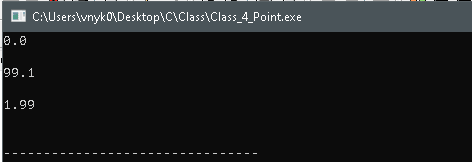
p.display();

point p2(1,99);

p2.display();

  return 0;

 }



**5. Create a class ComplexNumber with data members real, imaginary. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class**

#include<iostream>

using namespace std;

class complex

{

int real,imaginary;

public:

complex()

{

real,imaginary=0;

}

complex(int real,int imaginary)

{

this->real=real;

this->imaginary=imaginary;

}

void setreal(int r)

{

this->real=r;

}

void setimaginary(int i)

{

this->imaginary=i;

}

int getreal()

{

return this->real;

}

int getimaginary()

{

return this->imaginary;

}

void display()

{

cout<<"real: "<<real<<"\nimaginary: "<<imaginary<<endl<<endl;

}

};

int main()

{

complex c1;

c1.display();

c1.setreal(90);

c1.setimaginary(3);

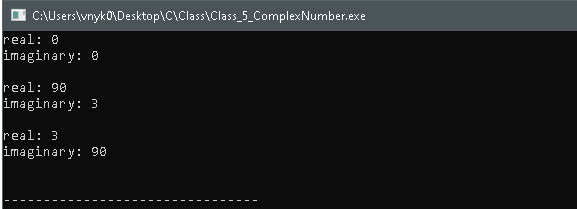
c1.display();

complex c2(3,90);

c2.display();

return 0;

}



**---------C-Day1 Functions------**

**1.Finding F from C (temp).**

#include<stdio.h>

float get();

float calculation(float);

int main()

{

printf("\nC to F : %.2f F",calculation(get()));

return 0;

}

float get()

{

float Celsius;

printf("\nEnter Temprature in Celsius: ");

scanf("%f",&Celsius);

return Celsius;

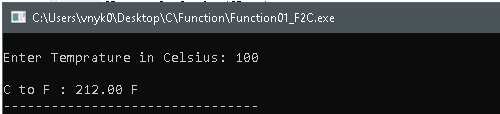
}

float calculation(float Celsius)

{

return Celsius\*(9.0/5.0)+32.0;

}



**2.Finding area and perimeter of rectangle or circle.**

 #include<stdio.h>

#define PI 3.14f

void Roperation(void);

void Coperation(void);

int main()

{

int ch=0;

while(ch!=3)

{

printf("\nChoose option:\n1. Rectangle's area and perimeter\n2. Circle's area and perimeter\n3. Exit\n\nYour selection was: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

Roperation();

break;

case 2:

Coperation();

break;

default:

printf("Bye..\n");

}

}

return 0;

}

void Roperation()

{

float l,b,area,peri;

printf("\nPlease enter length and breadth of rectangle: ");

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

area=l\*b;

peri=2\*(l+b);

printf("Area: %.2f\nPerimeter: %.2f\n---------------------------------------------------",area,peri);

}

void Coperation()

{

float rad,area, perm;

    printf("\nEnter radius of circle: ");

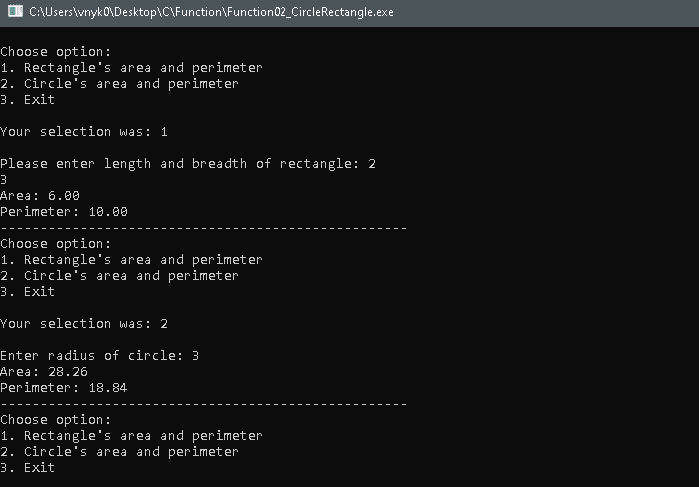
    scanf("%f",&rad);

    area=PI\*rad\*rad;

    perm=2\*PI\*rad;

    printf("Area: %.2f \nPerimeter: %.2f\n---------------------------------------------------",area,perm);

}



**3.Accept a 3 digit number from user and find the sum of the digits and also reverse the number**

 #include<stdio.h>

void sum\_rev(int);

int main()

{

int num;

printf("\nPlease enter 3 digit number: ");

scanf("%d",&num);

if(num<1000 && num>99)

sum\_rev(num);

else

printf("\nPlease enter 3 digit number only");

return 0;

}

void sum\_rev(int num)

{

int sum=0,rev=0,remain;

while(num)

{

remain=(num%10);

sum+=remain;

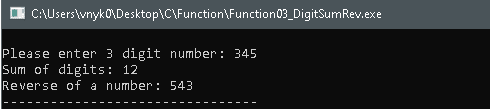
rev=rev\*10+remain;

num/=10;

}

printf("Sum of digits: %d\nReverse of a number: %d",sum,rev);

}



**4.Check if the given number is even or odd.**

 #include<stdio.h>

int get();

void check(int);

int main()

{

check(get());

return 0;

}

int get()

{

int num;

printf("\nPlease enter a number to check for even odd: ");

scanf("%d",&num);

return num;

}

void check(int num)

{

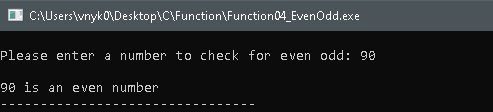
if(num%2==0)

printf("\n%d is an even number",num);

else

printf("\n&d is an odd number",num);

}



**5.Calculating total salary based on basic. If basic <=5000 da, ta and hra will be 10%,20% and 25% respectively  otherwise  da, ta and hra will be 15%,25% and 30% respective**

#include<stdio.h>

void calculate(float);

float get(void);

int main()

{

calculate(get());

return 0;

}

float get()

{

float basic;

printf("\nEnter basic salary: ");

scanf("%f",&basic);

return basic;

}

void calculate(float basic)

{

float da,hra,ta;

float total\_salary;

if(basic<=5000)

{

da  = basic \* 0.1;

        hra = basic \* 2.5;

        ta = basic \* 0.2;

}

else

{

da  = basic \* 1.5;

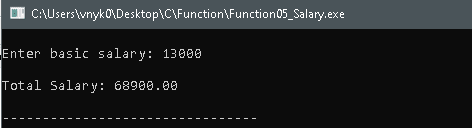
        hra = basic \* 0.3;

        ta = basic \* 2.5;

}

printf("\nTotal Salary: %.2f\n",total\_salary=basic+da+hra+ta);

}



**6.Find the price of item when discount is given (specify different discount based on price)**

 #include<stdio.h>

void get\_calculate();

void calculate(float,float);

int main()

{

get\_calculate();

return 0;

}

void get\_calculate()

{

float price,discount\_percentage;

printf("\nPlease enter price of the item : ");

scanf("%f",&price);

printf("\nPlease enter discount on the item : ");

scanf("%f",&discount\_percentage);

calculate(price,discount\_percentage);

}

void calculate(float price,float discount\_percentage)

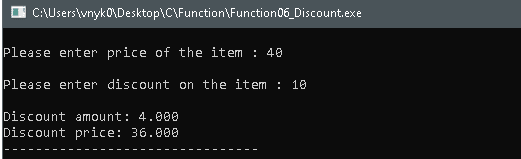
{

float discount\_amount,discount\_price;

printf("\nDiscount amount: %.3f",discount\_amount=discount\_percentage\*price/100);

printf("\nDiscount price: %.3f",discount\_price=price-discount\_amount);

}



**7.Write a program to find greatest of three numbers using nested if-else.**

#include<stdio.h>

void get\_calculation();

float calculation(float,float,float);

int main()

{

get\_calculation();

return 0;

}

void get\_calculation()

{

float a,b,c;

printf("\nEnter 3 numbers: ");

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

printf("\nGreatest of three number is: %.2f",calculation(a,b,c));

}

float calculation(float a,float b,float c)

{

//float greatest;

if(a>=b)

if(a>=c)

return a;

else

return b;

else

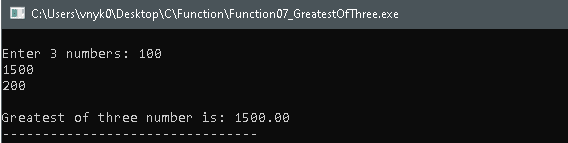
if(b>=c)

return b;

else

return c;

}



**8.Accept two numbers from user and an operator (+,-,/,\*,%) based on that perform the desired operations.**

 #include<stdio.h>

#include<stdlib.h>

float calculate(float,float,char);

int main()

{

float a,b;

char ch;

printf("\nEnter 2 numbers: ");

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

printf("\nOperation to be performed (+ - \* /) :");

scanf(" %c",&ch);

printf("%.2f %c %.2f = %.2f",a,ch,b,calculate(a,b,ch));

return 0;

}

float calculate(float a,float b,char ch)

{

float result;

switch(ch)

{

case '+':

result=a+b;

break;

case '-':

result=a-b;

break;

case '\*':

result=a\*b;

break;

case '/':

result=a/b;

break;

default:

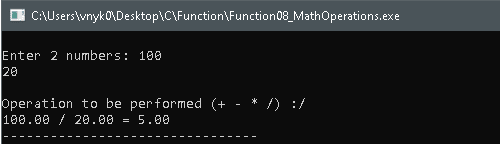
printf("\nWrong operation chosen");

exit(0);

}

return result;

}



**9.Display a menu to the user (like 1.Even Odd 2. Basic salary etc), ask the user to enter his choice, then based on that perform the desired operations.**

#include<stdio.h>

#include "Link.h"

void calling(int);

int main()

{

int ch;

printf("\nChoose option:\n1: fahrenheit to celsius\n2: Area, Perimeter of Rectangle and circle \n3: Sum, Reverse of 3 digit number \n4: Even or odd number\n5: Calculate total salary from basic salary\n6: Price of item from Discount\n7: Greatest of three numbers\n8: Mathematical Operation\n9: Discounted price for user \nYour choice will be : ");

scanf("%d",&ch);

calling(ch);

return 0;

}

void calling(int ch)

{

switch(ch){

case 1:

F2Cget();

break;

case 2:

RectCircleGet();

break;

case 3:

sum\_rev\_get();

break;

case 4:

EvenOddget();

break;

case 5:

SalaryGet();

break;

case 6:

Discountget();

break;

case 7:

Greatestget();

break;

case 8:

Mathgetput();

break;

case 9:

StudentDiscountget();

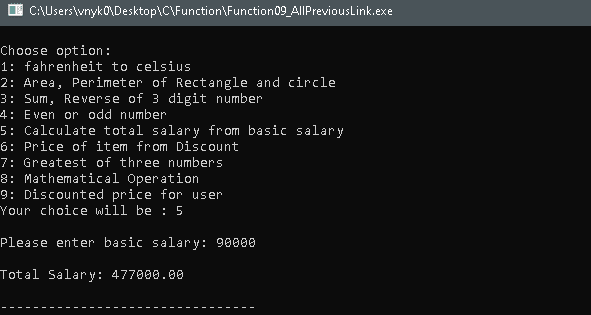
break;

default:

printf("\nWrong input entered. Please try again");

}

}



**10.Accept the price from user. Ask the user if he is a student (user may say yes or no). If he is a student and he has purchased more than 500 than discount is 20% otherwise discount is 10%. But if he is not a student then if he has purchased more than 600 discount is 15% otherwise there is not discount.**

#include<stdio.h>

#include<stdlib.h>

void StudentDiscountget();

void StudentDiscountcalculate(float,int);

int main()

{

StudentDiscountget();

return 0;

}

void StudentDiscountget()

{

int ch;

float price;

printf("\nEnter the price : ");

scanf("%f",&price);

printf("\nAre you a student\nEnter '1' for yes\nEnter '2 for no \n");

scanf("%d",&ch);

StudentDiscountcalculate(price,ch);

}

void StudentDiscountcalculate(float price,int ch)

{

float purchase;

switch(ch)

{

case 1:

if(price>500)

purchase=(price\*20)/100;

else

purchase=(price\*10)/100;

printf("Discounted Price is %.2f",(price-purchase));

break;

case 2:

if(price>600) {

purchase=(price\*15)/100;

printf("Discounted Price is %.2f",(price-purchase));

}

else

printf("Not eligible for discount\nYour Price=%.2f",price);

break;

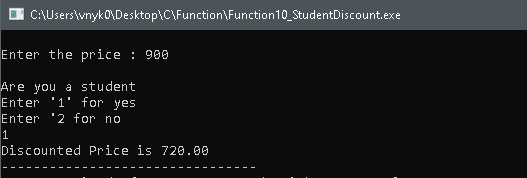
default:

printf("\nWrong input entered");

exit(0);

}

}



**11.Accept a number with 1 or 2 digit from user and display it in words.**

#include<stdio.h>

#include <string.h>

void get();

void convert(char\*);

int main()

{

get();

return 0;

}

void get(

{

char n[3];

printf("\nEnter a number: ");

scanf("%s",&n);

printf("");

convert(n);

}

void convert(char \*n)

{

int length=strlen(n);

if(length==0)

{

printf("Empty string");

return;

}

char \*singles[]={ "zero", "one", "two", "three", "four","five","six", "seven", "eight", "nine"};

char \*two\_digit[]={"", "ten", "eleven", "twelve", "thirteen", "fourteen","fifteen", "sixteen","seventeen", "eighteen", "nineteen"};

char \*tens\_digits[]={"", "", "twenty", "thirty", "forty", "fifty","sixty", "seventy", "eighty", "ninety"};

if(length==1)

{

printf("%s\n",singles[\*n-'0']);

return;

}

while(\*n!='\0')

{

if (\*n == '1')

{

int sum = \*n - '0' + \*(n + 1)- '0';

printf("%s\n", two\_digit[sum]);

return;

}

else if(\*n=='2' && \*(n+1)==0)

{

printf("Twenty\n");

return }

else

{

int i = \*n - '0';

printf("%s ", i? tens\_digits[i]: "");

++n;

if (\*n != '0')

printf("%s ", singles[\*n - '0']);

}

++n

}

}

