PROGRAM-52

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- Get the input of student: Name, Roll No, Marks in 6 subjects in 12th. Find if the student is eligible for admission in Delhi University. A student is eligible for DU if he has scored 95 % or more in Best 4.

**Code:**

#include<stdio.h>

struct student{

char \*name;

int roll\_no;

int marks[6];

};

int main(){

struct student s;

s.name=malloc(1024\*sizeof(char));

scanf("%s",s.name);

scanf("%d",&s.roll\_no);

for(int i=0;i<6;i++){

scanf("%d",&s.marks[i]);

}

for(int i=0;i<2;i++){

int min=s.marks[i];

int k = i;

for(int j=i;j<6;j++){

if(min>s.marks[j]){

k=j;

min = s.marks[j];

}

}

int t = s.marks[i];

s.marks[i] = s.marks[k];

s.marks[k] = t;

}

printf("\n");

float sum=0;

for(int i=2;i<6;i++){

sum = sum+s.marks[i];

}

float p = sum/4;

printf("Name of the student: %s\n",s.name);

printf("Roll number: %d\n",s.roll\_no);

printf("Percentage of best 4 subject: %0.2f\n",p);

if(p>=95){

printf("Status: Eligible");

}

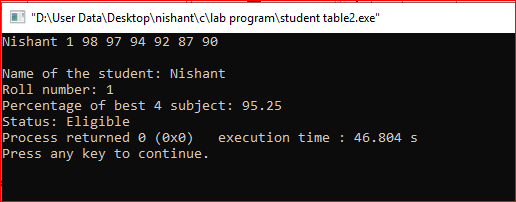
else{

printf("Status: Not Eligible");

}

}

**Output:**



PROGRAM-53

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- Write a program to store and print the roll no., name, age and marks of a student using structures. Write a program to store the roll no. (Starting from 1), name and age of 5 students and then print the details of the student with roll.

**Code:**

#include<stdio.h>

struct student{

int roll\_no;

char \*name;

int age;

int phy\_marks;

int chem\_marks;

int math\_marks;

};

int main(){

struct student s[5];

for(int i=0;i<5;i++){

scanf("%d",&s[i].roll\_no);

s[i].name = malloc(1024\*sizeof(int));

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

scanf("%d",&s[i].age);

scanf("%d",&s[i].phy\_marks);

scanf("%d",&s[i].chem\_marks);

scanf("%d",&s[i].math\_marks);

}

for(int i=0;i<5;i++){

printf("%d\t",s[i].roll\_no);

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

printf("%d\t",s[i].age);

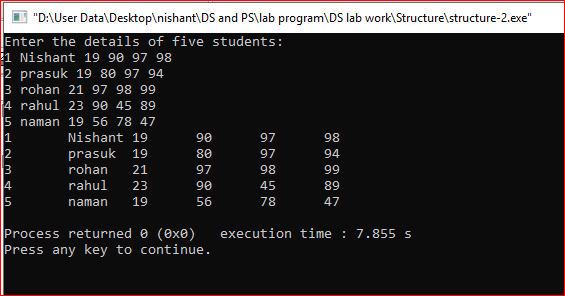
printf("%d\t",s[i].phy\_marks);

printf("%d\t",s[i].chem\_marks);

printf("%d\t",s[i].math\_marks);

}

}



PROGRAM-54

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- Write a program to store and print the roll no., name, age, address and marks of 15 students using structure.

**Code:**

#include<stdio.h>

struct student{

char \*name;

int roll\_no;

char section;

char \*address;

float percentage;

long mobile\_no;

};

int main(){

struct student s[15];

printf("Enter the details: \n");

for(int i=0;i<15;i++){

s[i].name = malloc(1024\*sizeof(char));

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

scanf("%d",&s[i].roll\_no);

scanf("%c",&s[i].section);

s[i].address = malloc(1024\*sizeof(char));

scanf("%s",s[i].address);

scanf("%f",&s[i].percentage);

scanf("%ld",&s[i].mobile\_no);

}

for(int i=0;i<15;i++){

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

printf("%d\t",s[i].roll\_no);

printf("%c\t",s[i].section);

printf("%s\t",s[i].address);

printf("%f\t",s[i].percentage);

printf("%ld",s[i].mobile\_no);

printf("\n");

}

return 0;

}

PROGRAM-55

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- Write a program to add two distances in inch-feet using structure. The values of the distances is to be taken from the user.

**Code:**

#include<stdio.h>

struct height{

int feet;

int inch;

};

int main(){

struct height h[2];

for(int i=0;i<2;i++){

printf("Enter height - %d: ",i+1);

scanf("%d",&h[i].feet);

scanf("%d",&h[i].inch);

}

int a=(h[0].inch+h[1].inch)/12;

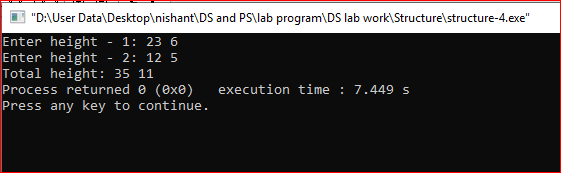
int b=(h[0].inch+h[1].inch)%12;

int c=(h[0].feet+h[1].feet)+a;

printf("Total height: %d %d",c,b);

}

**Output:**



PROGRAM-56

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- Write a program to add two complex numbers using structure. The values of the complex number is to be taken from the user.

**Code:**

#include<stdio.h>

struct complex{

int realPart;

int complexPart;

};

int main(){

struct complex c[2];

for(int i=0;i<2;i++){

printf("Enter the complex number-%d: ",i+1);

scanf("%d",&c[i].realPart);

scanf("%d",&c[i].complexPart);

}

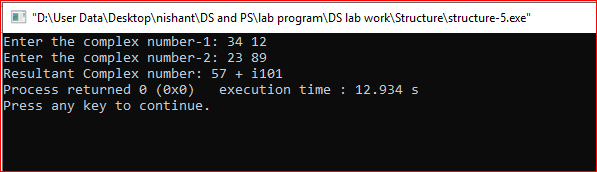
int a = c[0].realPart+c[1].realPart;

int b = c[0].complexPart+c[1].complexPart;

printf("Resultant Complex number: %d + i%d",a,b);

}

**Output:**



PROGRAM-57

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a program to add two time in hour, minute and second using structure. The values of the time is to be taken from the user.

**Code:**

#include<stdio.h>

struct time{

int hour;

int min;

int sec;

};

int main(){

struct time t[2];

for(int i=0;i<2;i++){

printf("Enter the time %d: ",i+1);

scanf("%d",&t[i].hour);

scanf("%d",&t[i].min);

scanf("%d",&t[i].sec);

}

int a = (t[0].sec+t[1].sec)/60;

int b = (t[0].sec+t[1].sec)%60;

int c = (t[0].min+t[1].min)/60+a;

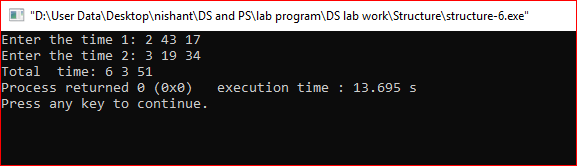
int d = ((t[0].sec+t[1].sec)+a)%60;

int e = (t[0].hour+t[1].hour)+c-a;

printf("Total time: %d %d %d",e,d,b);

}

**Output:**



PROGRAM-58

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Enter the marks of 5 students in Chemistry, Mathematics and Physics (each out of 100) using a structure named Marks having elements roll no., name, chem\_marks, maths\_marks and phy\_marks and then display the percentage

**Code:**

#include<stdio.h>

#include<stdlib.h>

struct student{

int roll\_no;

char \*name;

int age;

int phy\_marks;

int chem\_marks;

int math\_marks;

};

int main(){

struct student s[5];

printf("Enter the details: \n");

for(int i=0;i<5;i++){

scanf("%d",&s[i].roll\_no);

s[i].name = malloc(1024\*sizeof(int));

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

scanf("%d",&s[i].age);

scanf("%d",&s[i].phy\_marks);

scanf("%d",&s[i].chem\_marks);

scanf("%d",&s[i].math\_marks);

}

printf("\n---------------------------------\n");

for(int i=0;i<5;i++){

printf("%d\t",s[i].roll\_no);

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

printf("%d\t",s[i].age);

printf("%d\t",s[i].phy\_marks);

printf("%d\t",s[i].chem\_marks);

printf("%d\t",s[i].math\_marks);

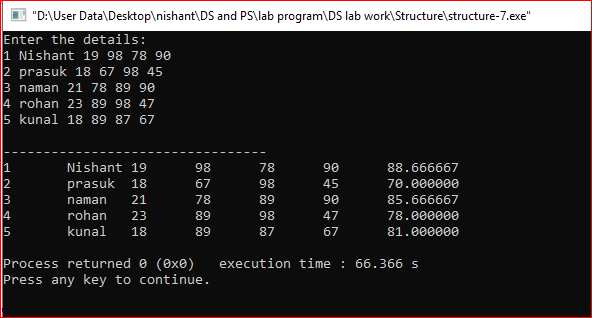
int sum=s[i].phy\_marks+s[i].chem\_marks+s[i].math\_marks;

printf("%f\n",sum/3.0);

}

}

**Output:**



PROGRAM-59

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a program to add, subtract and multiply two complex numbers using structures to function

**Code:**

#include<stdio.h>

struct complex{

int realPart;

int imaginaryPart;

};

int main(){

struct complex c[2];

for(int i=0;i<2;i++){

printf("Enter the complex number-%d: ",i+1);

scanf("%d",&c[i].realPart);

scanf("%d",&c[i].imaginaryPart);

}

int a = c[0].realPart+c[1].realPart;

int b = c[0].imaginaryPart+c[1].imaginaryPart;

printf("Add two complex number: %d + i%d\n",a,b);

int c1 = c[0].realPart-c[1].realPart;

int d = c[0].imaginaryPart-c[1].imaginaryPart;

printf("Subtract two complex number: %d + i(%d)\n",c1,d);

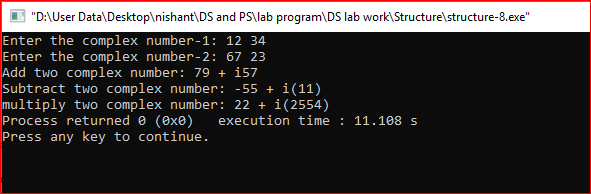
int e = (c[0].realPart\*c[1].realPart)-(c[0].imaginaryPart\*c[1].imaginaryPart);

int f = (c[0].realPart\*c[1].imaginaryPart)+(c[0].imaginaryPart\*c[1].realPart);

printf("multiply two complex number: %d + i(%d)",e,f);

}

**Output:**



PROGRAM-60

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a structure to store the roll no., name, age (Between 11 to 14) and address of students (more than 10). Store the information of the students.

- Write a function to print the names of all the students having age 14.

- Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).

**Code:**

#include<stdio.h>

struct student{

int roll\_no;

char \*name;

int age;

char \*address;

};

void displayByAge(struct student \*s,int n,int age){

printf("\nName of Students having age %d: \n",age);

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

if(s[i].age==age){

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

}

}

}

void displayEvenRoll\_no(struct student \*s,int n){

printf("\nName of Students having even roll\_no: \n");

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

if(s[i].roll\_no%2==0){

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

}

}

}

void displayByRoll\_no(struct student \*s,int n,int roll\_no){

printf("\nDetail of Student: \n");

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

if(s[i].roll\_no==roll\_no){

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

printf("%d\t",s[i].age);

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

break;

}

}

}

int main(){

struct student \*s;

int n;

printf("Enter the number of students: ");

scanf("%d",&n);

s=malloc(n\*sizeof(struct student));

printf("Enter the details of %d student:\n",n);

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

scanf("%d",&s[i].roll\_no);

s[i].name = malloc(25\*sizeof(char));

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

scanf("%d",&s[i].age);

s[i].address=malloc(100\*sizeof(char));

scanf("%s",s[i].address);

}

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

int age;

printf("Enter the Age: ");

scanf("%d",&age);

displayByAge(s,n,age);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

displayEvenRoll\_no(s,n);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

int roll\_no;

printf("\nEnter the Roll\_no: ");

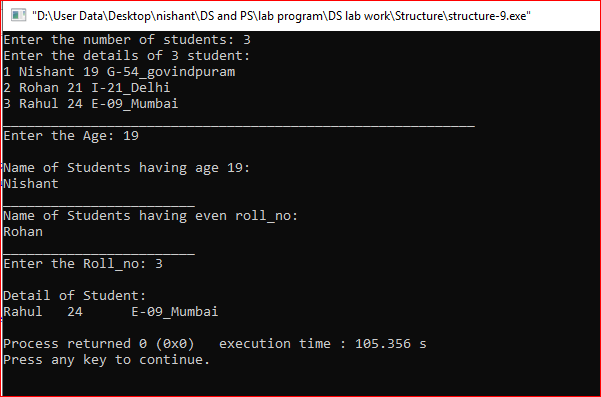
scanf("%d",&roll\_no);

displayByRoll\_no(s,n,roll\_no);

return 0;

}

**Output:**



PROGRAM-61

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a structure to store the name, account number and balance of customers (more than 10) and store their information.

1 - Write a function to print the names of all the customers having balance less than $200.

2 - Write a function to add $100 in the balance of all the customers having more than $1000 in their balance and then print the incremented value of their balance.

**Code:**

#include<stdio.h>

struct customer{

char \*name;

int account\_no;

int balance;

};

void displayByBalance(struct customer \*c,int n,int balance){

printf("\nName of customers having balance less than %d: \n",balance);

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

if(c[i].balance<balance){

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

}

}

}

void displayIncrementBy100(struct customer \*c,int n){

printf("\nAccount Number which is incremented by $100: \n");

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

if(c[i].balance>1000){

printf("%d\t",c[i].account\_no);

printf("%d\n",c[i].balance+100);

}

}

}

int main(){

struct customer \*c;

int n;

printf("Enter the number of customers: ");

scanf("%d",&n);

c=malloc(n\*sizeof(struct customer));

printf("Enter the details of %d customer:\n",n);

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

c[i].name = malloc(25\*sizeof(char));

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

scanf("%d",&c[i].account\_no);

scanf("%d",&c[i].balance);

}

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

int balance;

printf("Enter the Balance: ");

scanf("%d",&balance);

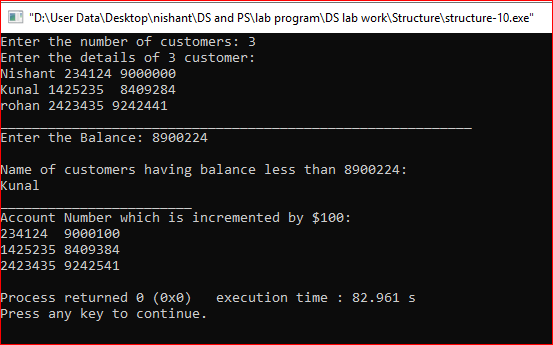
displayByBalance(c,n,balance);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

displayIncrementBy100(c,n);

}

**Output:**



PROGRAM-62

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a program to compare two dates entered by user. Make a structure named Date to store the elements day, month and year to store the dates. If the dates are equal, display "Dates are equal" otherwise display "Dates are not

**Code:**

#include<stdio.h>

struct date{

int day;

int month;

int year;

};

int main(){

struct date d[2];

for(int i=0;i<2;i++){

printf("Enter the date-%d: ",i+1);

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

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

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

}

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

if(d[0].day==d[1].day && d[0].month==d[1].month && d[0].year==d[1].year){

printf("\nDates are Equal");

}

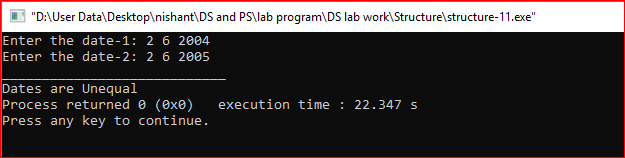
else{

printf("\nDates are Unequal");

}

}

**Output:**



PROGRAM-63

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Write a structure to store the names, salary and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries. Hours of work per day 8 Increase in salary $50

**Code:**

#include<stdio.h>

struct worker{

char \*name;

int salary;

int working\_hour;

};

void displayByWorking\_hour(struct worker \*w,int n){

printf("\nName of workers whose salary increased by $50: \n");

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

if(w[i].working\_hour>10){

printf("%s\t",w[i].name);

printf("%d\n",w[i].salary+50);

}

}

}

int main(){

struct worker \*w;

int n;

printf("Enter the number of workers: ");

scanf("%d",&n);

w=malloc(n\*sizeof(struct worker));

printf("Enter the details of %d worker:\n",n);

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

w[i].name = malloc(25\*sizeof(char));

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

scanf("%d",&w[i].salary);

scanf("%d",&w[i].working\_hour);

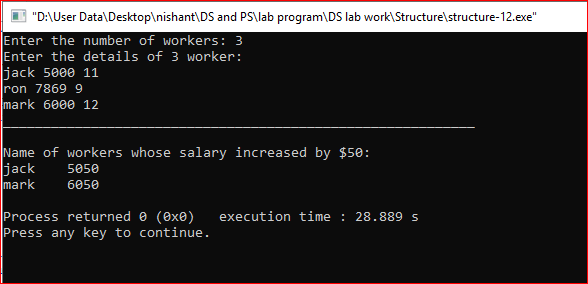
}

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

displayByWorking\_hour(w,n);

}

**Output:**



PROGRAM-64

**Program Name**- Structure **Name**- Nishant Kumar

**Domain-** Structure  **AdmissionNo**.2021B0101054

**Problem Statement**- . Let us work on the menu of a library. Create a structure containing book information like accession number, name of author, book title and flag to know whether book is issued or not. Create a menu in which the following can be:

1 - Display book information

2- Add a new book

3 - Display all the books in the library of a particular author

4 - Display the number of books of a particular title

5 - Display the total number of books in the library

6 - Issue a book

(If we issue a book, then its number gets decreased by 1 and if we add a book, its number gets increased by 1)

**Code:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct library{

int accession\_number;

char name\_of\_author[25];

int book\_title[25];

int flag;

};

void Input(struct library \*ptr,int n){

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

scanf("%d %s %s %d",&ptr[i].accession\_number,&ptr[i].name\_of\_author,&ptr[i].book\_title,&ptr[i].flag);

}

}

void Output(struct library \*ptr,int n){

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

printf("%d\t%s\t%s\t%d\n",ptr[i].accession\_number,ptr[i].name\_of\_author,ptr[i].book\_title,ptr[i].flag);

}

printf("\n");

}

void displayInfo(struct library \*ptr,int n){

char s[25];

printf("Enter the Title of the book: ");

scanf("%s",&s);

int i=0;

while(i<n){

if(strcmp(s,ptr[i].book\_title)==0){

printf("%d\t%s\t%d\n",ptr[i].accession\_number,ptr[i].name\_of\_author,ptr[i].flag);

}

i++;

}

printf("\n");

}

void addBook(struct library \*ptr,int n){

printf("Enter the information: \n");

ptr = realloc(ptr,1\*sizeof(struct library));

scanf("%d %s %s %d",&ptr[n].accession\_number,&ptr[n].name\_of\_author,&ptr[n].book\_title,&ptr[n].flag);

printf("\n");

}

void displayByAuthor(struct library \*ptr,int n){

char s[25];

printf("Enter the Author Name: ");

scanf("%s",&s);

int i=0;

while(i<n){

if(strcmp(s,ptr[i].name\_of\_author)==0){

printf("%d\t%s\t%d\n",ptr[i].accession\_number,ptr[i].book\_title,ptr[i].flag);

}

i++;

}

printf("\n");

}

void displayNumByTitle(struct library \*ptr,int n){

char s[25];

printf("Enter the Title of the book: ");

scanf("%s",&s);

int i=0,c=0;

while(i<n){

if(strcmp(s,ptr[i].book\_title)==0){

c++;

}

i++;

}

printf("%d\n",c);

printf("\n");

}

void issueBook(struct library \*ptr,int n){

char s[25];

printf("Enter the Title of the book: ");

scanf("%s",&s);

int i=0;

while(i<n){

if(strcmp(s,ptr[i].book\_title)==0){

ptr[i].flag=1;

break;

}

i++;

}

printf("Book Issued successfully.\n");

printf("\n");

}

int main(){

int n;

struct library \*p;

printf("==========Library==========\n");

printf("Enter the number of books: ");

scanf("%d",&n);

printf("Entry(accession number, name of author, book title and flag)\n");

p = (struct library \*)malloc(n\*sizeof(struct library));

Input(p,n);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Enter 1 To Display the information of Book\n");

printf("Enter 2 To Add a new book\n");

printf("Enter 3 To Display all the books in the library of a particular author.\n");

printf("Enter 4 To Display number of books of particular title.\n");

printf("Enter 5 To Display the information of all books in library.\n");

printf("Enter 6 To issue a book:\n");

printf("Enter 7 to Exit library\n");

while(1){

int option;

printf("Enter your option: ");

scanf("%d",&option);

switch (option){

case 1:

displayInfo(p,n);

break;

case 2:

addBook(p,n);

n++;

break;

case 3:

displayByAuthor(p,n);

break;

case 4:

displayNumByTitle(p,n);

break;

case 5:

Output(p,n);

break;

case 6:

issueBook(p,n);

break;

case 7:

exit(1);

default: printf("Invalid Option.");

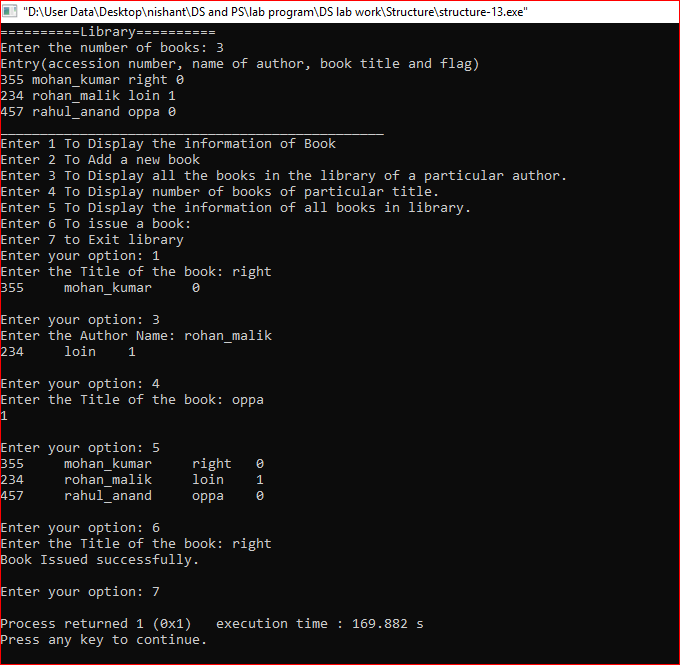
}

}

return 0;

}

**Output:**



PROGRAM-65

**Program Name**- Number conversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Decimal to Binary Conversion

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define size 100

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

int x;

printf("Enter the number: ");

scanf("%d",&x);

initialise();

while(x!=0){

int rem =x%2;

push(rem);

x=x/2;

}

while(!isempty()){

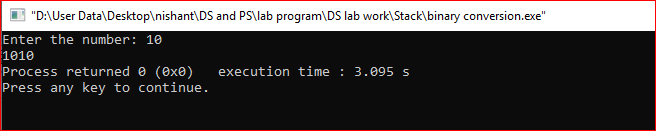
int a= pop();

printf("%d",a);

}

}

**Output:**



PROGRAM-66

**Program Name**- Number conversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Decimal to Octal Conversion

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define size 100

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

int x;

printf("Enter the number: ");

scanf("%d",&x);

initialise();

while(x!=0){

int rem =x%8;

push(rem);

x=x/8;

}

while(!isempty()){

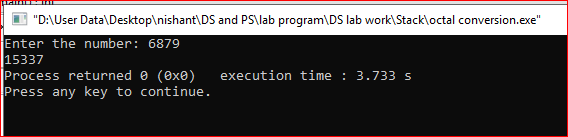
int a= pop();

printf("%d",a);

}

}

**Output:**



PROGRAM-67

**Program Name**- Number conversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Decimal to Hexadecimal Conversion

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define size 100

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

int x;

printf("Enter the number: ");

scanf("%d",&x);

initialise();

while(x!=0){

int rem =x%16;

push(rem);

x=x/16;

}

while(!isempty()){

int a= pop();

if(a>=10){

printf("%c",65+(a-10));

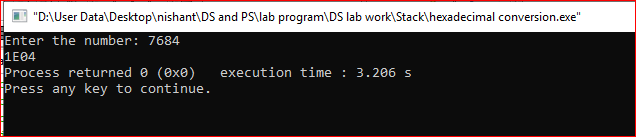
}

else printf("%d",a);

}

}

**Output:**



PROGRAM-68

**Program Name**- Number conversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Decimal to Any Base Conversion

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define size 100

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

int x,n;

printf("Enter the number: ");

scanf("%d",&x);

printf("Enter the base: ");

scanf("%d",&n);

initialise();

while(x!=0){

int rem =x%n;

push(rem);

x=x/n;

}

while(!isempty()){

int a= pop();

if(a>=10){

printf("%c",65+(a-10));

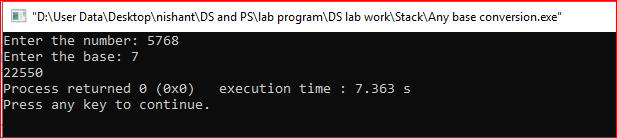
}

else printf("%d",a);

}

}

**Output:**



PROGRAM-71

**Program Name**- Palindrome check **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program to check if the given string is a palindrome using stacks

**Code:**

#include<stdio.h>

#include<string.h>

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

char s1[20],s2[20];

int i=0;

printf("Enter the String: ");

gets(s1);

while(s1[i]!='\0'){

push(s1[i]);

i++;

}

int j=0;

while(!isempty()){

s2[j]=pop();

j++;

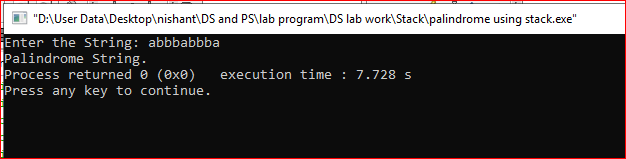
}

if(strcmp(s1,s2)==0) printf("Palindrome String.");

else printf("Not a Palindrome.");

}

**Output:**



PROGRAM-72

**Program Name**- string reverse **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program to reverse the given String using Stack

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define size 100

#define TURE 1

#define FALSE 0

#include"header files/mystack.h"

int main(){

char s[20];

int i=0;

printf("Enter the String: ");

gets(s);

while(s[i]!='\0'){

push(s[i]);

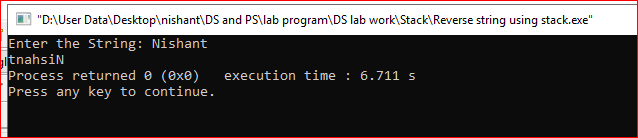
i++;

}

while(!isempty()) printf("%c",pop());

}

**Output:**



PROGRAM-73

**Program Name**- Expression interconversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Postfix Evaluation

**Code:**

#include<stdio.h>

#include<string.h>

#include"header files/myStack.h"

int Evaluate(int a,int b,char op){

switch(op){

case '+': return a+b;

case '\*': return a\*b;

case '-': return a-b;

case '/': return a/b;

}

}

int PostfixEvalution(char \*exp){

initialise();

char symb;

int a,b,val,i=0;

while(exp[i]!='\0'){

symb=exp[i];

if(symb>='0'&&symb<='9'){

push(symb-48);

}

else{

b= pop();

a= pop();

val =Evaluate(a,b,symb);

push(val);

}

i++;

}

int x=pop();

return x;

}

int main(){

char exp[20];

printf("Enter the Expression(postFix): ");

gets(exp);

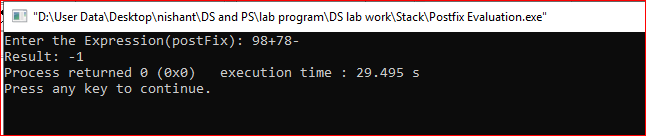
int x=PostfixEvalution(exp);

printf("Result: %d",x);

return 0;

}

**Output:**



PROGRAM-74

**Program Name**- Expression interconversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Prefix Evaluation

**Code:**

#include<stdio.h>

#define STACKSIZE 10

#define TRUE 1

#define FALSE 0

#include"header files/myStack.h"

int Evaluate(int a,int b,char op){

switch(op){

case '+': return a+b;

case '\*': return a\*b;

case '-': return a-b;

case '/': return a/b;

}

}

int main(){

char PS[20],symb;

int val,a,b,x,i=0;

gets(PS);

strrev(PS);

while(PS[i]!='\0'){

symb=PS[i];

if(symb>='0'&&symb<='9'){

push(symb-48);

}

else{

a= pop();

b= pop();

val =Evaluate(a,b,symb);

push(val);

}

i++;

}

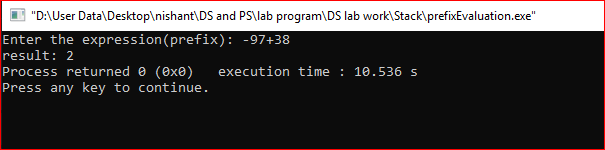
x=pop();

printf("%d",x);

return 0;

}

**Output:**



PROGRAM-75

**Program Name**- Expression interconversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- . Program for Infix to Postfix Conversion

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TRUE 1

#define FALSE 0

#include"header files/myStack.h"

int prcd(char a,char b){

if(a=='^' || a=='\*' || a=='/' || a=='%'){

if(b=='^'){

return FALSE;

}

else{

return TRUE;

}

}

else{

if(b=='+' || b=='-'){

return TRUE;

}

else{

return FALSE;

}

}

}

void infixtopostfix(char infix[]){

int i=0,p=0;

char postfix[20];

char x,symbol;

initialise();

while(infix[i]!='\0'){

symbol=infix[i];

i++;

if((symbol>='a' && symbol<='z')||(symbol>='A' && symbol<='Z')||(symbol>='0' && symbol<='9')){

postfix[p]=symbol;

p++;

}

else{

while(!isempty() && prcd(stacktop(),symbol)){

x=pop();

postfix[p]=x;

p++;

}

push(symbol);

}

}

while(!isempty()){

x=pop();

postfix[p]=x;

p++;

}

postfix[p]='\0';

for(int i=0;postfix[i]!='\0';i++){

if(postfix[i]!='(' && postfix[i]!=')')

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

}

}

int main(){

char infix[20];

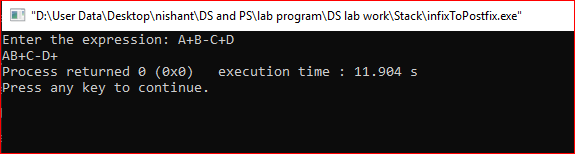
printf("Enter the expression: ");

gets(infix);

infixtopostfix(infix);

}

**Output:**



PROGRAM-76

**Program Name**- Expression interconversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Infix to Prefix Conversion

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TRUE 1

#define FALSE 0

#include"header files/myStack.h"

int prcd(char a,char b){

if(a=='^' || a=='\*' || a=='/' || a=='%'){

if(b=='^'){

return FALSE;

}

else{

return TRUE;

}

}

else{

if(b=='+' || b=='-'){

return TRUE;

}

else{

return FALSE;

}

}

}

void infixToprefix(char infix[]){

int i=0,p=0;

char prefix[20];

char x,symbol;

initialise();

strrev(infix);

while(infix[i]!='\0'){

symbol=infix[i];

i++;

if(symbol>='a' && symbol<='z'){

prefix[p]=symbol;

p++;

}

else{

while(!isempty() && !prcd(symbol,stacktop())){

x=pop();

prefix[p]=x;

p++;

}

push(symbol);

}

}

while(!isempty()){

x=pop();

prefix[p]=x;

p++;

}

prefix[p]='\0';

strrev(prefix);

printf("%s",prefix);

}

int main(){

char infix[20];

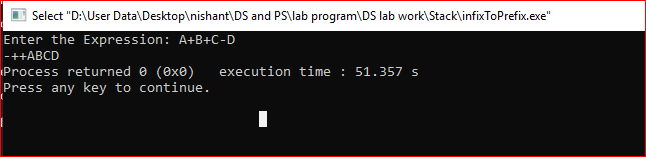
printf("Enter the Expression: ");

gets(infix);

infixToprefix(infix);

}

**Output:**



PROGRAM-77

**Program Name**- Expression interconversion **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Infix to Postfix Conversion (Infix Expression with Parenthesis)

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TRUE 1

#define FALSE 0

#include"header files/myStackPointer.h"

int prcd(char a,char b){

if(a=='('){

return FALSE;

}

else if(a=='^' || a=='\*' || a=='/' || a=='%'){

if(b=='^' || b=='('){

return FALSE;

}

else{

return TRUE;

}

}

else{

if(b=='+' || b=='-' || b==')'){

return TRUE;

}

else{

return FALSE;

}

}

}

void infixtopostfix(char infix[]){

int i=0,p=0;

char postfix[20];

char x,symbol;

struct stack opstack;

initialise(&opstack);

while(infix[i]!='\0'){

symbol=infix[i];

i++;

if((symbol>='a' && symbol<='z')){

postfix[p]=symbol;

p++;

}

else{

while(!isempty(&opstack) && prcd(stacktop(&opstack),symbol)){

x=pop(&opstack);

postfix[p]=x;

p++;

}

if(symbol==')'){

pop(&opstack);

}

else push(&opstack,symbol);

}

}

while(!isempty(&opstack)){

x=pop(&opstack);

postfix[p]=x;

p++;

}

postfix[p]='\0';

for(int i=0;postfix[i]!='\0';i++){

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

}

}

int main(){

char infix[20];

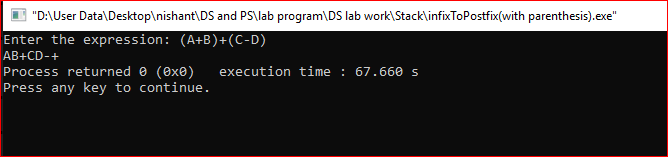
printf("Enter the expression: ");

gets(infix);

infixtopostfix(infix);

}

**Output:**



PROGRAM-79

**Program Name**- Multi stack **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for implementation of 2 stacks using a single Array

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define TRUE 1

#define FALSE 0

struct twoStack{

int \*arr;

int size;

int \*TOP;

};

void intailise(struct twoStack \*ms,int n){

ms->arr=calloc(n,sizeof(int));

ms->size=n/2;

ms->TOP=calloc(ms->size,sizeof(int));

int j=-1;

for(int i=0;i<ms->size;i++){

ms->TOP[i]=j;

j=j+(ms->size);

}

}

int isEmpty(struct twoStack \*ms,int n){

int s=ms->size;

int a=-1+(n-1)\*s;

if(ms->TOP[n-1]==a) return TRUE;

else return FALSE;

}

void push(struct twoStack \*ms,int n,int val){

int s=ms->size;

int a=((n)\*s)-1;

if(ms->TOP[n-1]==a){

printf("Stack:%d Overflow",n);

exit(1);

}

ms->TOP[n-1]++;

ms->arr[ms->TOP[n-1]]=val;

}

int pop(struct twoStack \*ms,int n){

if(isEmpty(ms,n)){

printf("Stack:%d Underflow",n);

exit(1);

}

int x=ms->arr[ms->TOP[n-1]];

ms->TOP[n-1]--;

return x;

}

int stackTop(struct twoStack \*ms,int n){

int a=ms->TOP[n-1];

return ms->arr[a];

}

int main(){

struct twoStack ms;

intailise(&ms,16);

push(&ms,1,10);

push(&ms,1,20);

push(&ms,1,20);

push(&ms,2,40);

push(&ms,2,50);

push(&ms,2,60);

pop(&ms,1);

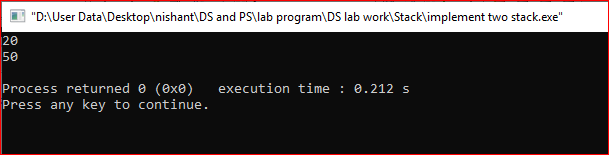
pop(&ms,2);

printf("%d\n",stackTop(&ms,1));

printf("%d\n",stackTop(&ms,2));

}

**Output:**



PROGRAM-80

**Program Name**- Maximum/Minimum **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Finding Maximum/Minimum in the Stack

**Code:**

#include<stdio.h>

#include"header files/myStackPointer.h"

int main(){

int n,y;

struct stack fs;

struct stack ss;

initialise(&fs);

initialise(&ss);

printf("Enter the number of element: ");

scanf("%d",&n);

int i=0;

while(i<n){

int x;

scanf("%d",&x);

push(&fs,x);

i++;

}

int max = stacktop(&fs);

while(!isempty(&fs)){

y = stacktop(&fs);

if(y>max){

max = y;

}

else{

pop(&fs);

push(&ss,y);

}

}

while(!isempty(&ss)){

pop(&ss);

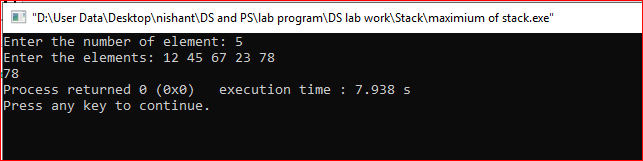
push(&fs,y);

}

printf("%d",max);

}

**Output:**



**Code:**

#include<stdio.h>

#include"header files/myStackPointer.h"

int main(){

int n,y;

struct stack fs;

struct stack ss;

initialise(&fs);

initialise(&ss);

printf("Enter the number of element: ");

scanf("%d",&n);

int i=0;

while(i<n){

int x;

scanf("%d",&x);

push(&fs,x);

i++;

}

int min = stacktop(&fs);

while(!isempty(&fs)){

y = stacktop(&fs);

if(y<min){

min = y;

}

else{

pop(&fs);

push(&ss,y);

}

}

while(!isempty(&ss)){

pop(&ss);

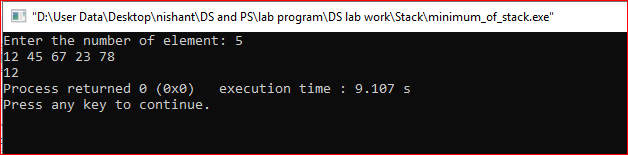
push(&fs,y);

}

printf("%d",min);

}

**Output:**



PROGRAM-81

**Program Name**- Sorting **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Sorting of stack

**Code:**

#include<stdio.h>

#include"header files/myStackPointer.h"

int main(){

int x,n;

struct stack input;

struct stack temp;

initialise(&input);

initialise(&temp);

printf("Enter the number of element: ");

scanf("%d",&n);

printf("Enter the element: ");

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

scanf("%d",&x);

push(&input,x);

}

int a;

while(!isempty(&input)){

if(isempty(&temp)){

a = pop(&input);

push(&temp,x);

}

else{

a = pop(&input);

while(a<stacktop(&temp) && !isempty(&temp)){

push(&input,pop(&temp));

}

push(&temp,a);

}

}

while(!isempty(&temp)){

push(&input,pop(&temp));

}

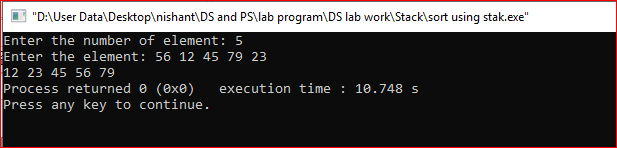
for(int i=0;i<n;i++){

printf("%d ",pop(&input));

}

}

**Output:**



PROGRAM-82

**Program Name**- Multi stack **Name**- Nishant Kumar

**Domain-** Stack **AdmissionNo**.2021B0101054

**Problem Statement**- Program for implementation of multiple stack in one Array

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define TRUE 1

#define FALSE 0

struct multi\_Stack{

int \*arr;

int size;

int \*TOP;

};

void intailise(struct multi\_Stack \*ms,int n,int m){

ms->arr=calloc(n,sizeof(int));

ms->size=n/m;

ms->TOP=calloc(ms->size,sizeof(int));

int j=-1;

for(int i=0;i<ms->size;i++){

ms->TOP[i]=j;

j=j+(ms->size);

}

}

int isEmpty(struct multi\_Stack \*ms,int n){

int s=ms->size;

int a=-1+(n-1)\*s;

if(ms->TOP[n-1]==a) return TRUE;

else return FALSE;

}

void push(struct multi\_Stack \*ms,int n,int val){

int s=ms->size;

int a=((n)\*s)-1;

if(ms->TOP[n-1]==a){

printf("Stack:%d Overflow",n);

exit(1);

}

ms->TOP[n-1]++;

ms->arr[ms->TOP[n-1]]=val;

}

int pop(struct multi\_Stack \*ms,int n){

if(isEmpty(ms,n)){

printf("Stack:%d Underflow",n);

exit(1);

}

int x=ms->arr[ms->TOP[n-1]];

ms->TOP[n-1]--;

return x;

}

int main(){

struct multi\_Stack ms;

int m;

intailise(&ms,16,4);

}

PROGRAM-83

**Program Name**- Liner Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program of Array Implementation of Linear Queue

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define QUEUESIZE 10

#define TRUE 1

#define FALSE 0

struct Queue{

int item[QUEUESIZE];

int Front;

int Rear;

};

void initialise(struct Queue \*q){

q->Front=0;

q->Rear=-1;

}

int isEmpty(struct Queue \*q){

if(q->Rear-q->Front+1==0){

return TRUE;

}

else{

return FALSE;

}

}

void enqueue(struct Queue \*q,int x){

if(q->Rear==QUEUESIZE-1){

printf("Queue overflow");

exit(1);

}

q->Rear=q->Rear+1;

q->item[q->Rear]=x;

}

**int dequeue(struct Queue \*q){**

**if (isEmpty(q)){**

**printf("Queue underflow");**

**exit(1);**

**}**

**int x=q->item[q->Front];**

**q->Front=q->Front+1;**

**return x;**

**}**

PROGRAM-84

**Program Name**- Circular Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program of Array Implementation of Circular Queue

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define QUEUESIZE 10

#define TRUE 1

#define FALSE 0

struct Queue{

int item[QUEUESIZE];

int Front;

int Rear;

};

void initialise(struct Queue \*cq){

cq->Front=QUEUESIZE-1;

cq->Rear=QUEUESIZE-1;

}

int isEmpty(struct Queue \*cq){

if(cq->Rear == cq->Front){

return TRUE;

}

else{

return FALSE;

}

}

void enqueue(struct Queue \*cq,int x){

if((cq->Rear+1)%QUEUESIZE==cq->Front){

printf("Queue overflow");

exit(1);

}

cq->Rear=(cq->Rear+1)%QUEUESIZE;

cq->item[cq->Rear]=x;

}

int dequeue(struct Queue \*cq){

if (isEmpty(cq)){

printf("Queue underflow");

exit(1);

}

cq->Front=(cq->Front+1)%QUEUESIZE;

int x=cq->item[cq->Front];

return x;

}

PROGRAM-85

**Program Name**- DEQUE **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Array Implementation of Double Ended Queue

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define QUEUESIZE 20

#define TRUE 1

#define FALSE 0

struct Queue{

int item[QUEUESIZE];

int Front;

int Rear;

};

void initialise(struct Queue \*q){

q->Front=0;

q->Rear=-1;

}

int isEmpty(struct Queue \*q){

if(q->Rear-q->Front+1==0){

return TRUE;

}

else{

return FALSE;

}

}

void insertFront(struct Queue \*q,int x){

if(q->Rear==QUEUESIZE-1){

printf("Queue overflow");

exit(1);

}

q->Rear++;

q->item[q->Rear]=x;

}

void insertLast(struct Queue \*q,int x){

for(int i=q->Rear;i>=q->Front;q--){

q->item[i+1]=q->item[i];

}

q->Rear++;

q->item[q->Front]=x;

}

int main(){

struct Queue q;

initialise(&q);

insertFront(&q,1);

insertFront(&q,2);

insertFront(&q,3);

insertFront(&q,4);

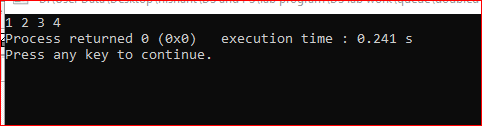
for(int i=0;i<4;i++){

printf("%d ",q.item[i]);

}

}

**Output:**



PROGRAM-86

**Program Name**- Priority Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Array Implementation of Priority Queue (Ascending Array)

**Code:**

#include<stdio.h>

int n;

void display(int \*a){

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

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

}

}

void arrayinsert(int \*a, int i, int x){

for(int j=n-1; j>=i; j--){

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

}

a[i]=x;

n=n+1;

}

void pqinsert(int \*a, int x){

int i=0;

while(i<n && x>=a[i]){

i=i+1;

}

arrayinsert(a, i, x);

}

void arraydelete(int \*a, int i){

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

a[j-1]=a[j];

}

n=n-1;

}

int pqdelete(int \*a){

int x;

x=a[0];

arraydelete(a,0);

return x;

}

int main(){

int a[10];

int n=0;

pqinsert(a,10);

pqinsert(a,20);

pqinsert(a,5);

pqinsert(a,25);

pqinsert(a,30);

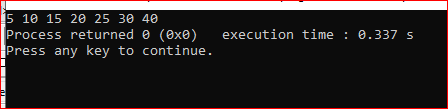
pqinsert(a,15);

pqinsert(a,40);

display(a);

}

**Output:**



PROGRAM-87

**Program Name**- Priority Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Array Implementation of Priority Queue (Descending Array)

**Code:**

#include<stdio.h>

int n;

void display(int \*a){

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

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

}

}

void arrayinsert(int \*a, int i, int x){

for(int j=n-1; j>=i; j--){

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

}

a[i]=x;

n=n+1;

}

void pqinsert(int \*a, int x){

int i=0;

while(i<n && x<=a[i]){

i=i+1;

}

arrayinsert(a, i, x);

}

void arraydelete(int \*a, int i){

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

a[j-1]=a[j];

}

n=n-1;

}

int pqdelete(int \*a){

int x;

x=a[0];

arraydelete(a,0);

return x;

}

int main(){

int a[10];

int n=0;

pqinsert(a,10);

pqinsert(a,20);

pqinsert(a,5);

pqinsert(a,25);

pqinsert(a,30);

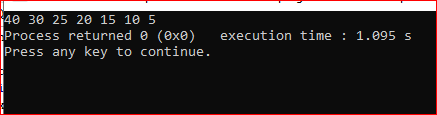
pqinsert(a,15);

pqinsert(a,40);

display(a);

}

**Output:**



PROGRAM-88

**Program Name**- Priority Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Heap Implementation of Priority Queue

**Code:**

#include<stdio.h>

int n;

void pqinsert(int a[], int data){

a[n]= data;

int x=n;

int y=(n-1)/2;

while(x>0 && a[x]<a[y]){

int temp=a[x];

a[x]=a[y];

a[y]= temp;

x=y;

y=(x-1)/2;

}

n=n+1;

}

void minheapify(int a[], int i){

while(2\*i+1<=n-1){

int l= 2\*i+1;

int r= 2\*i+1;

int min=l;

if(r<=n-1){

if(a[r]<a[l]){

min=r;

}

}

if(a[min]<a[i]){

int temp=a[min];

a[min]= a[i];

a[i]= temp;

}

else{

break;

}

i=min;

}

}

int pqdelete(int a[]){

int x=a[0];

a[0]= a[n-1];

n=n-1;

minheapify(a,0);

return x;

}

int main(){

int a[10];

pqinsert(a,10);

pqinsert(a,15);

pqinsert(a,20);

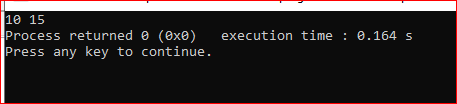
pqinsert(a,25);

printf("%d ",pqdelete(a));

printf("%d ",pqdelete(a));

}

**Output:**



PROGRAM-89

**Program Name**- Priority Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Stack implementation using Queue

**Code:**

#include<stdio.h>

#include"header files/Queue.h"

struct Queue q1,q2;

void push(int x){

enqueue(&q2,x);

while(!isEmpty(&q1)){

enqueue(&q2,dequeue(&q1));

}

while(!isEmpty(&q2)){

enqueue(&q1,dequeue(&q2));

}

}

int pop(){

if(isEmpty(&q1)){

printf("Invalid");

exit(1);

}

return dequeue(&q1);

}

int main(){

initialise(&q1);

initialise(&q2);

push(10);

push(20);

push(30);

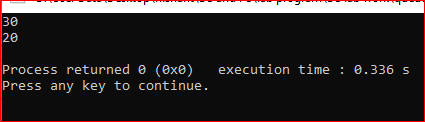
printf("%d\n",pop());

printf("%d\n",pop());

return 0;

}

**Output:**



PROGRAM-90

**Program Name**- Priority Queue **Name**- Nishant Kumar

**Domain-** Queue **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Queue implementation using Stack

**Code:**

#include<stdio.h>

#include<stdlib.h>

#include"header files/myStackPointer.h"

struct stack s1,s2;

void enQueue(int x){

while(!isempty(&s1)){

push(&s2,stacktop(&s1));

pop(&s1);

}

push(&s1,x);

while(!isempty(&s2)){

push(&s1,stacktop(&s2));

pop(&s2);

}

}

int deQueue(){

if(isempty(&s1)){

printf("Invalid");

exit(1);

}

int x=stacktop(&s1);

pop(&s1);

return x;

}

int main(){

initialise(&s1);

initialise(&s2);

enQueue(10);

enQueue(20);

enQueue(30);

enQueue(40);

enQueue(50);

enQueue(60);

printf("%d ",deQueue());

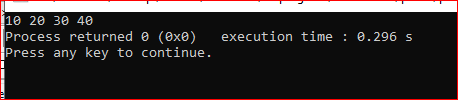
printf("%d ",deQueue());

printf("%d ",deQueue());

printf("%d ",deQueue());

}

**Output:**



PROGRAM-91

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Linear Linked List Primitive operations

**Code:**

#include<stdio.h>

#include<stdlib.h>

#include"Header files/LinkedList.h"

int main(){

struct Node \*START,\*p;

START = NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,30);

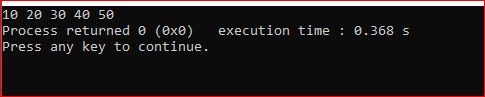
InsEnd(&START,40);

InsEnd(&START,50);

Traverse(&START);

}

**Output:**



PROGRAM-92

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for creation of Linked List header file and test of basic functions through that

**Code:**

#include<stdio.h>

#include<stdlib.h>

struct Node{

int info;

struct Node \*Next;

};

struct Node \*GetNode(){

struct Node \*p;

p=(struct Node\*)malloc(sizeof(struct Node));

return p;

};

void Traverse(struct Node \*\*START){

struct Node \*p;

p=\*START;

while(p!=NULL){

printf("%d ",p->info);

p=p->Next;

}

}

int CountNode(struct Node \*\*START){

int c=0;

struct Node \*q;

q=(\*START);

while(q!=NULL){

c++;

q=q->Next;

}

return c;

}

void InsBeg(struct Node \*\*START,int x){

struct Node \*p;

p=GetNode();

p->info=x;

p->Next=(\*START);

(\*START)=p;

}

void InsAft(struct Node \*\*q,int x){

struct Node \*p,\*r;

p=GetNode();

p->info=x;

r=(\*q)->Next;

p->Next=r;

(\*q)->Next=p;

}

void InsEnd(struct Node \*\*START,int x){

struct Node \*q,\*p;

q=(\*START);

if(q==NULL){

InsBeg(START,x);

}

else{

while(q->Next!=NULL){

q=q->Next;

}

p=GetNode();

p->info=x;

p->Next=NULL;

q->Next=p;

}

}

void Insert(struct Node \*\*START,int x,int n){

int a = CountNode(START);

if(n==1) InsBeg(START,x);

else if(n==a+1) InsEnd(START,x);

else if(n>1 && n<=a){

struct Node \*p;

p=\*START;

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

p=p->Next;

}

InsAft(&p,x);

}

else printf("Invalid Index");

}

void OrderedInsertion(struct Node \*\*START,int x){

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p!=NULL && x>p->info){

q=p;

p=p->Next;

}

if(q==NULL){

InsBeg(START,x);

}

else InsAft(&q,x);

}

void PQInsertion(struct Node \*\*START,int x){

OrderedInsertion(START,x);

}

int DelBeg(struct Node \*\*START){

int x;

struct Node \*p;

p=\*START;

x=p->info;

\*START=(\*START)->Next;

free(p);

return x;

}

int DelEnd(struct Node \*\*START){

int x;

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p->Next!=NULL){

q=p;

p=p->Next;

}

q->Next=NULL;

x=p->info;

free(p);

return x;

}

int DelAft(struct Node \*\*p){

int x;

struct Node \*q,\*r;

q=(\*p)->Next;

r=q->Next;

(\*p)->Next=r;

x=q->info;

free(p);

return x;

}

int PQdel(struct Node \*\*START){

int x = DelBeg(START);

}

struct Node\* middleElement(struct Node \*\*START){

struct Node \*T,\*R;

T=\*START;

R=(\*START)->Next;

while(R!=NULL && R->Next!=NULL){

T=T->Next;

R=R->Next;

R=R->Next;

}

return T;

}

PROGRAM-93

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for finding count of Nodes in Linked List.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

int countNode(struct Node \*\*START){

int c=0;

struct Node \*q;

q=(\*START);

while(q!=NULL){

c++;

q=q->Next;

}

return c;

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

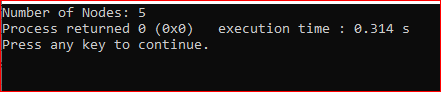
InsEnd(&START,4);

InsEnd(&START,5);

printf("Number of Nodes: %d",countNode(&START));

}

**Output:**



PROGRAM-94

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for concatenation of Linear Linked List.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void concat(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p;

p = \*START1;

while(p->Next!=NULL){

p=p->Next;

}

p->Next = \*START2;

}

int main(){

struct Node \*START1,\*START2;

START1 = NULL;

InsEnd(&START1,1);

InsEnd(&START1,2);

InsEnd(&START1,3);

InsEnd(&START1,4);

InsEnd(&START1,5);

START2 = NULL;

InsEnd(&START2,6);

InsEnd(&START2,7);

InsEnd(&START2,8);

InsEnd(&START2,9);

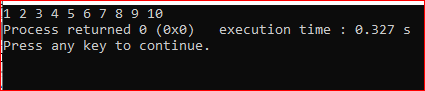
InsEnd(&START2,10);

concat(&START1,&START2);

Traverse(&START1);

}

**Output:**



PROGRAM-95

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program to implement linear search.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void LinearSearch(struct Node \*\*START,int key){

struct Node \*p;

p=\*START;

int t=0,c=0;

while(p!=NULL){

c++;

if(p->info==key){

t=1;

printf("%d is present at %d Node.",key,c);

break;

}

p=p->Next;

}

if(t==0) printf("Element is not present.");

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

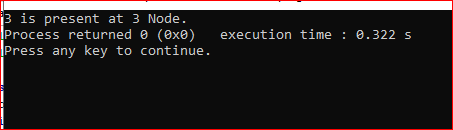
InsEnd(&START,4);

InsEnd(&START,5);

LinearSearch(&START,3);

}

**Output:**



PROGRAM-96

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program to insert an item at any given position in the linked List.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void insert(struct Node \*\*START,int x,int n){

int a = CountNode(START);

if(n==1) InsBeg(START,x);

else if(n==a+1) InsEnd(START,x);

else if(n>1 && n<=a){

struct Node \*p;

p=\*START;

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

p=p->Next;

}

InsAft(&p,x);

}

else printf("Invalid Index");

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,40);

InsEnd(&START,50);

InsEnd(&START,60);

Traverse(&START);

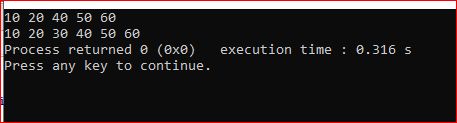
insert(&START,30,2);

printf("\n");

Traverse(&START);

}

**Output:**



PROGRAM-97

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Creation of Copy of the Linked list.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void Copy(struct Node \*\*START){

struct Node \*p,\*q;

q=NULL;

p=\*START;

while(p!=NULL){

InsEnd(&q,p->info);

p=p->Next;

}

Traverse(&q);

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

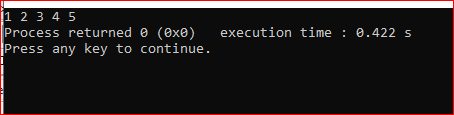
InsEnd(&START,4);

InsEnd(&START,5);

Copy(&START);

}

**Output:**



PROGRAM-98

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for counting nodes containing even and odd information.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void CountEvenOdd(struct Node \*\*START){

int c1=0,c2=0;

struct Node \*q;

q=(\*START);

while(q!=NULL){

if((q->info)%2==0){

c1++;

}

else c2++;

q=q->Next;

}

printf("Even = %d \nOdd = %d",c1,c2);

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

InsEnd(&START,4);

InsEnd(&START,5);

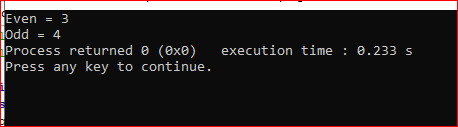
InsEnd(&START,6);

InsEnd(&START,7);

CountEvenOdd(&START);

}

**Output:**



PROGRAM-99

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Splitting a Linked List (in-place).

**Code:**

#include<stdio.h>

#include"LinkedList.h"

struct Node\* MiddleElement(struct Node \*\*START){

struct Node \*T,\*R;

T=\*START;

R=(\*START)->Next;

while(R!=NULL && R->Next!=NULL){

T=T->Next;

R=R->Next;

R=R->Next;

}

return T;

}

void Split(struct Node \*\*START){

struct Node \*START2,\*p;

p=(\*START);

START2=NULL;

p=MiddleElement(START);

START2 = p->Next;

p->Next = NULL;

Traverse(START);

printf("\n");

Traverse(&START2);

}

int main(){

struct Node \*START;

START=NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,30);

InsEnd(&START,40);

InsEnd(&START,50);

InsEnd(&START,60);

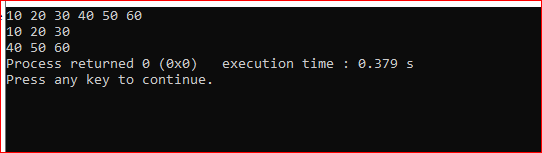
Traverse(&START);

printf("\n");

Split(&START);

}

**Output:**



PROGRAM-100

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Creation of Ascending Order Linear Linked List.

**Code:**

#include<stdio.h>

#include"header files/LinkedList.h"

void OrderedInsertion(struct Node \*\*START,int x){

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p!=NULL && x>p->info){

q=p;

p=p->Next;

}

if(q==NULL){

InsBeg(START,x);

}

else InsAft(&q,x);

}

int main(){

struct Node \*START;

START=NULL;

OrderedInsertion(&START,35);

OrderedInsertion(&START,40);

OrderedInsertion(&START,10);

OrderedInsertion(&START,25);

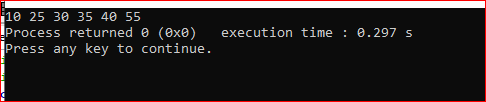
OrderedInsertion(&START,30);

OrderedInsertion(&START,55);

Traverse(&START);

}

**Output:**



PROGRAM-101

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Merging two sorted Linked List/unsorted link list.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void Merge(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p,\*q,\*START;

START=NULL;

p=\*START1;

q=\*START2;

while(p!=NULL && q!=NULL){

if(p->info<q->info){

InsEnd(&START,p->info);

p=p->Next;

}

else{

InsEnd(&START,q->info);

q=q->Next;

}

}

while(p!=NULL){

InsEnd(&START,p->info);

p=p->Next;

}

while(q!=NULL){

InsEnd(&START,q->info);

q=q->Next;

}

Traverse(&START);

}

int main(){

struct Node \*START1,\*START2;

START1=NULL;

START2=NULL;

int arr1[5] = {20,40,30,10,5};

int arr2[6] = {35,34,12,20,30,23};

for(int i=0;i<5;i++){

OrderedInsertion(&START1,arr1[i]);

}

for(int i=0;i<6;i++){

OrderedInsertion(&START2,arr2[i]);

}

Traverse(&START1);

printf("\n");

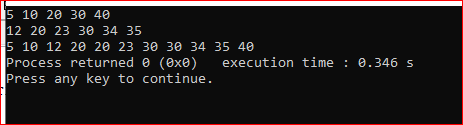
Traverse(&START2);

printf("\n");

Merge(&START1,&START2);

}

**Output:**



PROGRAM-102

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Union of two sorted Linked List (consider lists as a sets).

.

**Code:**

#include<stdio.h>

#include"LinkedList.h"

void UnionLL(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p,\*q,\*START;

START=NULL;

p=\*START1;

q=\*START2;

while(p!=NULL && q!=NULL){

if(p->info<q->info){

InsEnd(&START,p->info);

p=p->Next;

}

else{

if(p->info==q->info){

InsEnd(&START,p->info);

q=q->Next;

p=p->Next;

}

else{

InsEnd(&START,q->info);

q=q->Next;

}

}

}

while(p!=NULL){

InsEnd(&START,p->info);

p=p->Next;

}

while(q!=NULL){

InsEnd(&START,q->info);

q=q->Next;

}

Traverse(&START);

}

int main(){

struct Node \*START1,\*START2;

START1=NULL;

START2=NULL;

int arr1[5] = {20,40,30,10,5};

int arr2[6] = {35,34,12,20,30,23};

for(int i=0;i<5;i++){

OrderedInsertion(&START1,arr1[i]);

}

for(int i=0;i<6;i++){

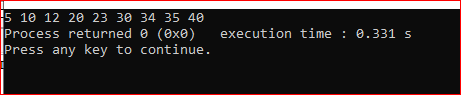
OrderedInsertion(&START2,arr2[i]);

}

UnionLL(&START1,&START2);

}

**Output:**



PROGRAM-103

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for intersection of two sorted Linked List (consider lists as a sets).

.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void IntersectionLL(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p,\*q,\*START;

START=NULL;

p=\*START1;

q=\*START2;

while(p!=NULL && q!=NULL){

if(p->info<q->info){

p=p->Next;

}

else{

if(p->info==q->info){

InsEnd(&START,p->info);

q=q->Next;

p=p->Next;

}

else{

q=q->Next;

}

}

}

Traverse(&START);

}

int main(){

struct Node \*START1,\*START2;

START1=NULL;

START2=NULL;

int arr1[5] = {20,40,30,10,5};

int arr2[6] = {35,34,12,20,30,23};

for(int i=0;i<5;i++){

OrderedInsertion(&START1,arr1[i]);

}

for(int i=0;i<6;i++){

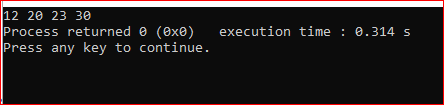
OrderedInsertion(&START2,arr2[i]);

}

IntersectionLL(&START1,&START2);

}

**Output:**



PROGRAM-104

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for difference of two sorted Linked List (consider lists as a sets).

.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void differenceLL(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p,\*q,\*START;

START=NULL;

p=\*START1;

q=\*START2;

while(p!=NULL && q!=NULL){

if(p->info<q->info){

InsEnd(&START,p->info);

p=p->Next;

}

else{

if(p->info==q->info){

q=q->Next;

p=p->Next;

}

else{

q=q->Next;

}

}

}

while(p!=NULL){

InsEnd(&START,p->info);

p=p->Next;

}

Traverse(&START);

}

int main(){

struct Node \*START1,\*START2;

START1=NULL;

START2=NULL;

int arr1[5] = {20,40,30,10,5};

int arr2[6] = {35,34,12,20,30,23};

for(int i=0;i<5;i++){

OrderedInsertion(&START1,arr1[i]);

}

for(int i=0;i<6;i++){

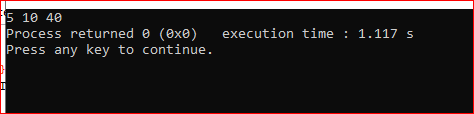
OrderedInsertion(&START2,arr2[i]);

}

differenceLL(&START1,&START2);

}

**Output:**



PROGRAM-105

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for symmetric difference of two sorted Linked List (consider lists as a sets).

.

**Code:**

#include<stdio.h>

#include"LinkedList.h"

void symmetricDifferenceLL(struct Node \*\*START1,struct Node \*\*START2){

struct Node \*p,\*q,\*START;

START=NULL;

p=\*START1;

q=\*START2;

while(p!=NULL && q!=NULL){

if(p->info<q->info){

InsEnd(&START,p->info);

p=p->Next;

}

else{

if(p->info==q->info){

q=q->Next;

p=p->Next;

}

else{

InsEnd(&START,q->info);

q=q->Next;

}

}

}

while(p!=NULL){

InsEnd(&START,p->info);

p=p->Next;

}

while(q!=NULL){

InsEnd(&START,q->info);

q=q->Next;

}

Traverse(&START);

}

int main(){

struct Node \*START1,\*START2;

START1=NULL;

START2=NULL;

int arr1[5] = {20,40,30,10,5};

int arr2[6] = {35,34,12,20,30,23};

for(int i=0;i<5;i++){

OrderedInsertion(&START1,arr1[i]);

}

for(int i=0;i<6;i++){

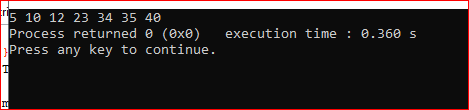
OrderedInsertion(&START2,arr2[i]);

}

symmetricDifferenceLL(&START1,&START2);

}

**Output:**



PROGRAM-106

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Finding the Middle element of a singly linked list in one pass

.

**Code:**

#include<stdio.h>

#include"header files/LinkedList.h"

struct Node\* middleElement(struct Node \*\*START){

struct Node \*T,\*R;

T=\*START;

R=(\*START)->Next;

while(R!=NULL && R->Next!=NULL){

T=T->Next;

R=R->Next;

R=R->Next;

}

return T;

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,30);

InsEnd(&START,40);

InsEnd(&START,50);

InsEnd(&START,60);

InsEnd(&START,70);

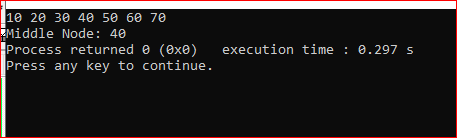
Traverse(&START);

struct Node \*m=middleElement(&START);

printf("\nMiddle Node: %d",m->info);

}

**Output:**



PROGRAM-107

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program to perform Binary Search on the Linked List.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

struct Node\* binarySearch(struct Node \*\*START,int key){

struct Node \*mid;

while((\*START)->Next!=NULL){

mid=middleElement(START);

if(mid->info==key) return mid;

else{

if(key<mid->info) mid->Next=NULL;

else (\*START) = mid->Next;

}

}

if((\*START)->info==key) return START;

else return NULL;

}

int main(){

struct Node \*START,\*p;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

InsEnd(&START,4);

InsEnd(&START,5);

Traverse(&START);

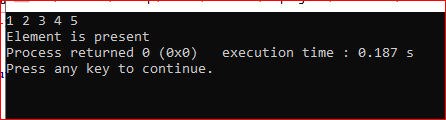
p=binarySearch(&START,8);

if(p!=NULL) printf("\nElement is present");

else printf("\nElement is not present");

}

**Output:**



PROGRAM-108

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Reversing the Linear Linked List (inplace and outplace).

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

#include"Header files/mystackPointer.h"

void ReverseInplace(struct Node \*\*START){

struct Node \*C,\*P,\*N;

C = \*START;

P = NULL;

N = C->Next;

while(C!=NULL){

C->Next = P;

P = C;

C = N;

if(N!=NULL) N = N->Next;

}

\*START = P;

}

void ReverseUsingStack(struct Node \*\*START){

struct Node \*p,\*START2;

START2=NULL;

struct stack s;

initialise(&s);

p=\*START;

while(p!=NULL){

push(&s,p->info);

p=p->Next;

}

while(!isempty(&s)){

InsEnd(&START2,pop(&s));

}

Traverse(&START2);

}

void Reverse(struct Node \*\*START){

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p!=NULL){

InsBeg(&q,p->info);

p=p->Next;

}

Traverse(&q);

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

InsEnd(&START,4);

InsEnd(&START,5);

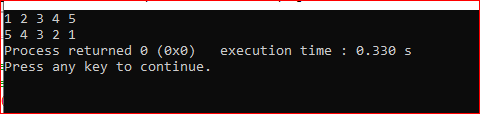
Traverse(&START);

printf("\n");

Reverse(&START);

}

**Output:**



PROGRAM-109

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program to print Linked List contents in reverse order.

**Code:**

void Reverse(struct Node \*\*START){

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p!=NULL){

InsBeg(&q,p->info);

p=p->Next;

}

Traverse(&q);

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

InsEnd(&START,4);

InsEnd(&START,5);

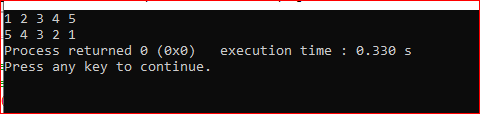
Traverse(&START);

printf("\n");

Reverse(&START);

**}**

**Output:**



PROGRAM-110

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Pair wise swap of elements in linked list.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void Swap(struct Node \*\*START){

struct Node \*p,\*q;

p=(\*START);

q=p->Next;

while(q!=NULL){

int t=p->info;

p->info = q->info;

q->info = t;

p=p->Next->Next;

q=q->Next;

if(q!=NULL){

q=q->Next;

}

}

}

int main(){

struct Node \*START;

START=NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,30);

InsEnd(&START,40);

InsEnd(&START,50);

InsEnd(&START,60);

Traverse(&START);

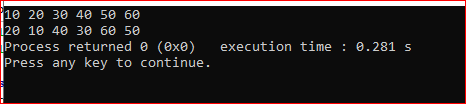
printf("\n");

Swap(&START);

Traverse(&START);

}

**Output:**



PROGRAM-111

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program to find kth node from the last in a single link list.

**Code:**

#include<stdio.h>

#include"header files/LinkedList.h"

int k\_NodeFromLast(struct Node \*\*START,int k){

int n=CountNode(START);

int m = n-k;

if(m<0){

printf("Invalid");

return -1;

}

int i=0;

while(i<m){

(\*START)=(\*START)->Next;

i++;

}

return \*START;

}

int main(){

struct Node \*START;

START=NULL;

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,30);

InsEnd(&START,40);

InsEnd(&START,50);

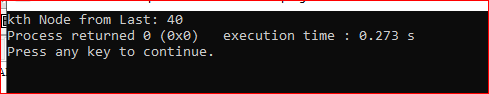
InsEnd(&START,60);

struct Node \*n=k\_NodeFromLast(&START,3);

printf("kth Node from Last: %d",n->info);

}

**Output:**



PROGRAM-112

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Sorting the Linear Linked List.

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void Sort(struct Node \*\*START){

struct Node \*p,\*q;

p=(\*START);

q=p->Next;

int f=0;

while(f!=1){

f=1;

while(q!=NULL){

if(p->info>q->info){

int t=p->info;

p->info = q->info;

q->info = t;

f=0;

}

p=p->Next;

q=q->Next;

}

}

}

int main(){

struct Node \*START;

START = NULL;

InsEnd(&START,30);

InsEnd(&START,40);

InsEnd(&START,10);

InsEnd(&START,20);

InsEnd(&START,50);

InsEnd(&START,70);

InsEnd(&START,60);

Traverse(&START);

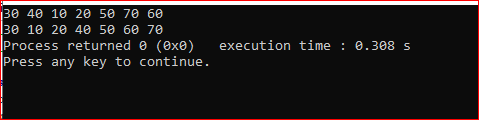
printf("\n");

Sort(&START);

Traverse(&START);

}

**Output:**



PROGRAM-113

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for finding if the given link list is palindrome or not

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

struct Node\* Reverse(struct Node \*\*START){

struct Node \*p,\*q;

p=\*START;

q=NULL;

while(p!=NULL){

InsBeg(&q,p->info);

p=p->Next;

}

return q;

}

int isPalindrome(struct Node \*\*START){

struct Node \*q,\*p;

p=(\*START);

q = Reverse(START);

while(p->Next!=NULL){

if(p->info!=q->info){

return 0;

}

p=p->Next;

q=q->Next;

}

return 1;

}

int main(){

struct Node \*START,\*p;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,2);

InsEnd(&START,3);

InsEnd(&START,3);

InsEnd(&START,1);

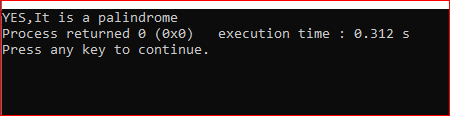
int x=isPalindrome(&START);

if(x==1) printf("YES");

else printf("NO");

}

**Output:**



PROGRAM-115

**Program Name**- Linear Linked List **Name**- Nishant Kumar

**Domain-** Linked List **AdmissionNo**.2021B0101054

**Problem Statement**- Program for Delete duplicate nodes in the Linked List

**Code:**

#include<stdio.h>

#include"Header files/LinkedList.h"

void duplicateElement(struct Node \*\*START){

struct Node \*p,\*q,\*r;

p=(\*START);

r=p;

q=p->Next;

while(p!=NULL){

while(q!=NULL){

if(q->info==p->info){

q=q->Next;

DelAft(&r);

}

else{

r=r->Next;

q=q->Next;

}

}

p=p->Next;

r=p;

if(p!=NULL) q=p->Next;

}

}

int main(){

struct Node \*START,\*p;

START = NULL;

InsEnd(&START,1);

InsEnd(&START,1);

InsEnd(&START,1);

InsEnd(&START,1);

InsEnd(&START,1);

InsEnd(&START,1);

InsEnd(&START,4);

InsEnd(&START,3);

InsEnd(&START,6);

InsEnd(&START,7);

Traverse(&START);

duplicateElement(&START);

printf("\n");

Traverse(&START);

}

**Output:**

