**PROGRAM NO. 1**

Program

1 - To insert an element at a given index

2 - To delete an element at a given index

**CODE**

/\*Program

1 - To insert an element at a given index

2 - To delete an element at a given index

\*/

#include<stdio.h>

#include<conio.h>

int size=5;

void insert(int \* ptr);

void del(int \* ptr);

void display(int \* ptr);

int main()

{ int arr[20],count,choice,flag=1;

clrscr();

printf("Enter %d elements in the array\n",size);

for(count=0;count<size;count++)

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

while(flag)

{printf("\n1 To insert an element in an array\n2 To delete an element in an array\n3 Exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1:insert(arr);

display(arr);

break;

case 2:del(arr);

display(arr);

break;

case 3:flag=0;

break;

default:printf("Enter correct choice\n");

}

}

getch();

return 0;

}

void display(int \* ptr)

{

int count;

printf("\nArray = ");

for(count=0;count<size;count++)

printf("%d ",ptr[count]);

printf("\n");

}

void insert(int \* ptr)

{

int pos,value,count;

printf("Enter the position to insert the element\n");

scanf("%d",&pos);

printf("Enter the value to insert\n");

scanf("%d",&value);

if(pos>0&&pos<size+2)

{

for(count=size+1;count>pos-1;count--)

ptr[count]=ptr[count-1];

ptr[pos-1]=value;

size=size+1;

}

else

printf("\nPlease enter valid position\n");

}

void del(int \* ptr)

{

int pos,count;

printf("Enter the position to delete\n");

scanf("%d",&pos);

if(pos>0&&pos<size+2)

{ for(count=pos;count<size;count++)

ptr[count-1]=ptr[count];

size=size-1;

}

else

printf("\nEnter a valid position\n");

}

**OUTPUT**

**Enter 5 elements in the array**

**5**

**2**

**4**

**9**

**8**

**1 To insert an element in an array**

**2 To delete an element in an array**

**3 Exit**

**1**

**Enter the position to insert the element**

**3**

**Enter the value to insert**

**78**

**Array = 5 2 78 4 9 8**

**1 To insert an element in an array**

**2 To delete an element in an array**

**3 Exit**

**2**

**Enter the position to delete**

**4**

**Array = 5 2 78 9 8**

**1 To insert an element in an array**

**2 To delete an element in an array**

**3 Exit**

**PROGRAM NO. 2**

Program to perform linear search in an array

**CODE**

/\*Program to perform linear search in an array\*/

#include<stdio.h>

#include<conio.h>

#define size 5

int main()

{int count,value,pos;

int ptr[size];

clrscr();

printf("\nEnter %d Elements in the array\n",size);

for(count=0;count<size;count++)

scanf("%d",&ptr[count]);

printf("\nEnter the element to be searched\n");

scanf("%d",&value);

for(count=0;count<size;count++)

{

if(ptr[count]==value)

{

pos=count+1;

break;

}

else

pos=-1;

}

if(pos>0)

printf("\nElement fount at position %d \n",pos);

else

printf("Element not found\n");

getch();

return 0;

}

**OUTPUT**

**Enter 5 Elements in the array**

**7**

**6**

**4**

**32**

**84**

**Enter the element to be searched**

**32**

**Element found at position 4**

**PROGRAM NO. 3**

Program to perform binary search on an array

**CODE**

/\*Program to perform binary search on an array\*/

#include<stdio.h>

#include<conio.h>

#define size 5

int main()

{int count,value,start,mid,last,pos=-1;

int ptr[size];

clrscr();

printf("Enter %d elements in array in increasing order\n",size);

for(count=0;count<size;count++)

scanf("%d",&ptr[count]);

printf("Enter element to be searched\n");

scanf("%d",&value);

start=0;

last=size-1;

mid=(start+last)/2;

while(start<=last)

{

if(ptr[mid]==value)

{

pos=mid+1;

break;

}

else

if(value>ptr[mid])

{

start=mid+1;

}

else

if(value<ptr[mid])

{

last=mid-1;

}

mid=(start+last)/2;

}

if(pos>0)

printf("Element found at position %d",pos);

else

printf("Element not found\n");

getch();

return 0;

}

**OUTPUT**

**Enter 5 elements in array in increasing order**

**1**

**6**

**8**

**23**

**45**

**Enter element to be searched**

**1**

**Element found at position 1**

**PROGRAM NO. 4**

Program to insert an element in an array after a given value and to delete an element from array

**CODE**

/\*Program to insert an element in an array after a given value and to delete an element from array \*/

#include<stdio.h>

#include<conio.h>

void display(int \* arr);

void delete(int \* arr);

void insert(int \* arr);

int size=5;

int main()

{ int value,count,choice;

int ptr[20];

clrscr();

printf("Enter %d elements in the array\n",size);

for(count=0;count<size;count++)

scanf("%d",&ptr[count]);

REP:printf("\n1 Delete an Element\n2 Insert an element\n");

scanf("%d",&choice);

switch(choice)

{

case 1: display(ptr);

delete(ptr);

printf("Updated array is");

display(ptr);

break;

case 2: display(ptr);

insert(ptr);

printf("Updated array is");

display(ptr);

break;

default:printf("\nEnter correct choice\n");

goto REP;

}

getch();

return 0;

}

void display(int \* arr)

{

int i;

printf("\nThe array is = ");

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

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

}

void delete(int \* arr)

{

int value,i,pos=-1;

printf("\nEnter the value to be deleted\n");

scanf("%d",&value);

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

if(arr[i]==value)

pos=i;

if(pos==-1)

printf("\nValue not found\n");

else

{

for(i=pos;i<size;i++)

arr[i]=arr[i+1];

size=size-1;

}

}

void insert(int \* arr)

{

int e\_value,value,temp,i,pos=-1;

printf("\nEnter the element after to insert the value\n");

scanf("%d",&e\_value);

printf("\nEnter the value to insert\n");

scanf("%d",&value);

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

if(arr[i]==e\_value)

pos=i+1;

if(pos==-1)

printf("\nElement not found\n");

else

{

size=size+1;

for(i=size;i>pos;i--)

arr[i]=arr[i-1];

arr[i]=value;

}

}

**OUTPUT**

**Enter 5 elements in the array**

**1**

**4**

**6**

**7**

**9**

**1 Delete an Element**

**2 Insert an element**

**3 Exit**

**1**

**The array is = 1 4 6 7 9**

**Enter the value to be deleted**

**1**

**The array is = 4 6 7 9**

**1 Delete an Element**

**2 Insert an element**

**3 Exit**

**2**

**The array is = 4 6 7 9**

**Enter the element after to insert the value**

**7**

**Enter the value to insert**

**3**

**The array is = 4 6 7 3 9**

**1 Delete an Element**

**2 Insert an element**

**3 Exit**

**PROGRAM NO. 5**

Program to perform selection sort using array

**CODE**

/\*Program to perform selection sort using array\*/

#include<stdio.h>

#include<conio.h>

#define size 5

void sort(int \* arr);

void display(int \* arr);

int main()

{ int ptr[size],count;

clrscr();

printf("Enter %d elements in the array\n",size);

for(count=0;count<size;count++)

scanf("%d",&ptr[count]);

sort(ptr);

printf("Sorted array is = ");

display(ptr);

getch();

return 0;

}

void sort(int \* arr)

{

int i,j,temp;

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

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

{

if(arr[j]<arr[i])

{

temp=arr[j];

arr[j]=arr[i];

arr[i]=temp;

}

}

}

void display(int \* arr)

{

int count;

for(count=0;count<size;count++)

printf("%d ",arr[count]);

}

**OUTPUT**

**Enter 5 elements in the array**

**5**

**12**

**9**

**147**

**2**

**Sorted array is = 2 5 9 12 147**

**PROGRAM NO. 6**

Program to store record of students using structure

**CODE**

/\*Program to store record of students using structure\*/

#include<stdio.h>

#include<conio.h>

#define size 2

struct student

{

int roll\_no;

char name[100];

char branch[50];

int avg;

}stu[size];

int main()

{ int count;

clrscr();

printf("Enter the details of %d students",size);

for(count=0;count<size;count++)

{

printf("\n\nDetails of student no. %d",(count+1));

printf("\nName = ");

gets(stu[count].name);

printf("Roll no = ");

scanf("%d",&stu[count].roll\_no);

printf("Avg marks = ");

scanf("%d",&stu[count].avg);

fflush(stdin);

printf("Branch name = ");

gets(stu[count].branch);

fflush(stdin);

}

printf("\nDetails of entered students are \n");

for(count=0;count<size;count++)

{

printf("\nName = %s",stu[count].name);

printf("\nRoll No = %d",stu[count].roll\_no);

printf("\nBranch = %s",stu[count].branch);

printf("\nAvg = %d\n",stu[count].avg);

}

getch();

return 0;

}

**OUTPUT**

**Enter the details of 2 students**

**Details of student no. 1**

**Name = Jhonny English**

**Roll no = 27**

**Avg marks = 99**

**Branch name = CSE**

**Details of student no. 2**

**Name = Prashant Chaudhary**

**Roll no = 89**

**Avg marks = 80**

**Branch name = CSE**

**Details of entered students are**

**Name = Jhonny English**

**Roll No = 27**

**Branch = CSE**

**Avg = 99.00**

**Name = Prashant Chaudhary**

**Roll No = 89**

**Branch = CSE**

**Avg = 80.00**

**PROGRAM NO. 7**

Program

1 - To insert an element in a linked list after a node

2 - To delete an element an element after a given node

3 - To delete a given node

**CODE**

/\*Program

1 - To insert an element in a linked list after a node

2 - To delete an element an element after a given node

3 - To delete a given node\*/

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<alloc.h>

struct list

{ int data;

struct list \* link;

}\*start;

typedef struct list linklist;

void create(linklist \* node);

void display(linklist \* node);

void insert(linklist \* node);

void del(linklist \* node);

void denode(linklist \* node);

int main()

{

int choice,flag=1;

clrscr();

start=(linklist \* ) malloc(sizeof(linklist));

create(start);

display(start);

if(start!=0)

while(flag==1)

{printf("\n1 To insert an element\n2 To delete after a given node\n3 To delete a given node\n4 Exit\n");

scanf("%d",&choice);

switch(choice)

{case 1:insert(start);

display(start);

break;

case 2:del(start);

display(start);

break;

case 3:denode(start);

display(start);

break;

case 4:flag=0;

break;

default:printf("Enter correct value\n");

}

if(start==0)

flag=0;

}

getch();

return 0;

}

void create(linklist \* node)

{ int flag=-1,a=0;

linklist \* next,\* temp;

while(flag<0)

{printf("Enter value in the list (Press 0 to stop)\n");

scanf("%d",&a);

if(a==0)

{if(node==start)

start=0;

else

temp->link=0;

flag=1;

}

else

{node->data=a;

next=(linklist \*)malloc(sizeof(linklist));

node->link=next;

temp=node;

node=next;

}

}

}

void display(linklist \* node)//To display the list

{

printf("\nThe list is = ");

if(node!=0)

while(node!=0)

{printf("%d",node->data);

if(node->link!=0)

printf("->");

node=node->link;

}

else

printf("EMPTY");

printf("\n");

}

void insert(linklist \*node)//To insert a node after a given element

{int value,flag=-1;

struct list \* pos,\*temp;

printf("\nEnter the value after to insert value\n");

scanf("%d",&value);

while(node!=0)

{if(node->data==value)

{pos=node;

flag=0;

break;

}

else

node=node->link;

}

if(flag==-1)

printf("\nValue entered not found\n");

else

{temp=(linklist \*)malloc(sizeof(linklist));

printf("\nEnter the value in the node\n");

scanf("%d",&temp->data);

temp->link=pos->link;

pos->link=temp;

}

}

void del(linklist \* node)//To delete a node after a given element

{int value,flag=-1;

linklist \* pos,\*temp;

printf("\nEnter the value after to delete\n");

scanf("%d",&value);

while(node!=0)

{if(node->data==value)

{pos=node;

flag=0;

break;

}

else

node=node->link;

}

if(flag==-1)

printf("\nValue entered not found\n");

else

{temp=pos->link;

pos->link=pos->link->link;

free(temp);

}

}

void denode(linklist \* node)//To delete a given node in the list

{

linklist \*temp,\* pos,\* ptr=node->link,\*pptr=node,\*loc,\*ploc;

int value,flag=-1;

printf("\nEnter the value to delete\n");

scanf("%d",&value);

if(start->data==value)

{ temp=start ;

start=start->link;

free(temp);

}

else

{

while(ptr!=0)

{if(ptr->data==value)

{ loc=ptr;

ploc=pptr;

flag=0;

break;

}

else

{ pptr=ptr;

ptr=ptr->link;

}

}

if(flag==-1)

printf("\nValue entered not found\n");

else

{ploc->link=loc->link;

free(loc);

}

}

}

**OUTPUT**

**Enter value in the list (Press 0 to stop)**

**3**

**Enter value in the list (Press 0 to stop)**

**6**

**Enter value in the list (Press 0 to stop)**

**9**

**Enter value in the list (Press 0 to stop)**

**0**

**The list is = 3->6->9**

**1 To insert an element**

**2 To delete after a given node**

**3 To delete a given node**

**4 Exit**

**1**

**Enter the value after to insert value**

**6**

**Enter the value in the node**

**7**

**The list is = 3->6->7->9**

**1 To insert an element**

**2 To delete after a given node**

**3 To delete a given node**

**4 Exit**

**2**

**Enter the value after to delete**

**3**

**The list is = 3->7->9**

**1 To insert an element**

**2 To delete after a given node**

**3 To delete a given node**

**4 Exit**

**3**

**Enter the value to delete**

**3**

**The list is = 7->9**

**1 To insert an element**

**2 To delete after a given node**

**3 To delete a given node**

**4 Exit**

**PROGRAM NO. 8**

Program for the impelementation of doubly linklist

**CODE**

/\* Program for the impelementation of doubly linklist\*/

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

#define null 0

struct dlist

{int data;

struct dlist \*link;

struct dlist \*prev;

} \* start,\*end;

void create();

void insert();

void del();

void display();

void main()

{int choice;

clrscr();

while(1)

{printf("\n enter your choice\n1-create\n2-insert\n3-delete\n4-display");

scanf("%d",&choice);

if(choice==1)

{create();

}

if(choice==2)

{insert();

}

if(choice==3)

{del();

}

if(choice==4)

{display();

}

if(choice==5)

{break;

}

}

getch();

}

void create()

{struct dlist \*ptr,\*cpt;

char ch;

ptr=(struct dlist\*)(malloc(sizeof(struct dlist)));

printf("enter the element ");

scanf("%d",&ptr->data);

ptr->prev=null;

start=ptr;

do

{cpt=(struct dlist\*)(malloc(sizeof(struct dlist)));

printf("enter the elements");

scanf("%d",&cpt->data);

ptr->link=cpt;

cpt->prev=ptr;

ptr=cpt;

printf("do you want to create more nodes press <y/n>");

ch=getche();

}while(ch=='y');

ptr->link=null;

}

void insert()

{struct dlist \*newptr;

newptr=(struct dlist\*)(malloc(sizeof(struct dlist)));

if(newptr==null)

{printf("overflow insertion not possible");

}

else

{printf("value to be inserted ");

scanf("%d",&newptr->data);

newptr->prev=null;

newptr->link=start;

start=newptr;

}

}

void del()

{struct dlist \*temp;

if(start==null)

{printf("underflow deletion not possible");

}

else

{temp=start;

printf("element being deleted is %d",start->data);

(start->link)->prev=null;

start=start->link;

free(temp);

}

}

void display()

{struct dlist \*ptr;

ptr=start;

printf("forwrd display");

while(ptr->link!=null)

{printf("\n%d",ptr->data);

ptr=ptr->link;

}

end=ptr;

{while(ptr!=null)

{printf("\n%d ",ptr->data);

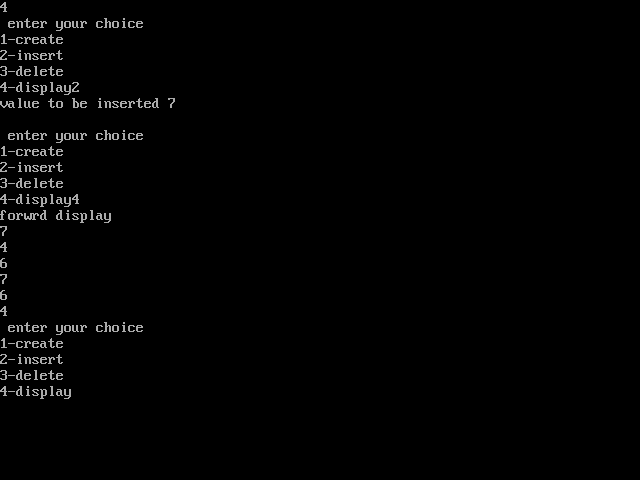
ptr=ptr->prev;

}

}

}

**Output**:



**PROGRAM NO. 9**

Program for array implementation of stack

**CODE**

/\* Program for array implementation of stack\*/

#include<stdio.h>

#include<conio.h>

#define maxsize 5

int stack[maxsize];

int top=0;

void push();

void pop();

void display();

void main()

{int choice;

clrscr();

while(1)

{printf("enter your choice \n1-push\n2-pop\n3-display");

scanf("%d",&choice);

if(choice==1)

{push();

}

if (choice==2)

{pop();

}

if(choice==3)

{display();

}

if(choice==4)

{break;

}

}

getch();

}

void push()

{ int item;

if(top==maxsize-1)

{printf("overflow condition and insertion not possible");

}

else

{printf("\nenter the item to be inserted");

scanf("%d",&item);

top++;

stack[top]=item;

}

}

void pop()

{int temp;

if(top==0)

{printf("\nunderflow condition and deletion not possible");

}

else

{temp=stack[top];

top--;

printf("\n poped item is %d",temp);

}

}

void display()

{int i;

if(top==0)

{printf("list is empty");

}

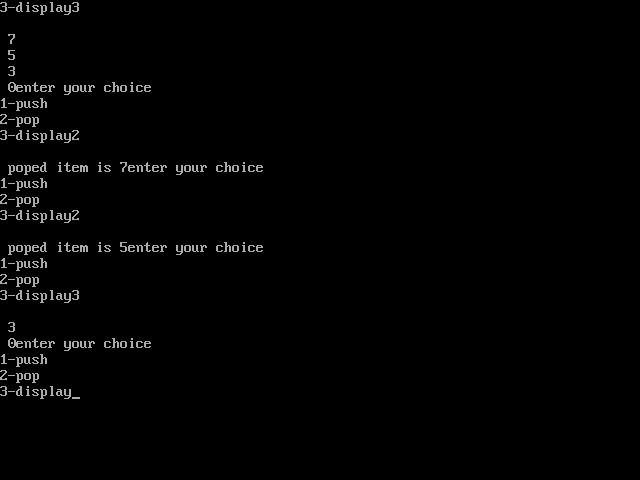
for(i=top;i>=0;i--)

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

}

}

**OUTPUT**



**PROGRAM NO. 10**

Program for the linklist implementation of stack

**CODE**

/\* Program for the linklist implementation of stack\*/

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

#define null 0

struct stack

{int data;

struct stack \*link;

}\*top;

void create();

void push();

void pop();

void display();

void main()

{int choice;

clrscr();

while(1)

{printf("\n enter your choice \n1-create\n2-pop\n3-push\n4-display");

scanf("%d",&choice);

if(choice==1)

{create();

}

if(choice==2)

{pop();

}

if(choice==3)

{push();

}

if(choice==4)

{display();

}

if( choice==5)

{break;

}

}

getch();

}

void create()

{struct stack \*ptr,\*cpt;

char ch;

ptr=(struct stack\*)(malloc(sizeof(struct stack)));

printf("\nenter element");

scanf("%d",&ptr->data);

ptr->link=null;

do

{cpt=(struct stack\*)(malloc(sizeof(struct stack)));

printf("\nenter the element");

scanf("%d",&cpt->data);

cpt->link=ptr;

ptr=cpt;

printf("\n do you want to create more nodes ");

ch=getche();

} while(ch=='y');

top=ptr;

}

void push()

{struct stack \*newptr;

newptr=(struct stack\*)(malloc(sizeof(struct stack)));

if( newptr==null)

{printf("\noverflow push not possible");

}

else

{printf("enter the element to be pushed");

scanf("%d",&newptr->data);

newptr->link=top;

top=newptr;

}

}

void pop()

{struct stack \*temp;

if(top==null)

{printf("\n underflow pop not possible");

}

else

{temp=top;

printf("poped value is %d",top->data);

top=top->link;

free(temp);

}

}

void display()

{struct stack \*ptr;

ptr=top;

while(ptr!=null)

{printf("\n%d",ptr->data);

ptr=ptr->link;

}

}

**OUTPUT**

