Stack Using Double Pointer

/* **Priyanshu Saini Roll.No.24** */

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
   int info;
   struct node *next;
}nodetype;

void push(nodetype **top,int val){
   nodetype *p = (nodetype *)malloc(sizeof(nodetype));
   if( p == NULL){
      printf("\n No memory allocated.");
   }else{
      p->info = val;
      p->next = *top;
      *top = p;
   }
}

void pop(nodetype **top){
   if(*top == NULL){
      printf("No element to pop.");
   }else{
      nodetype *temp = *top;
      printf("\n %d popped.\n",temp->info);
      *top = temp->next;
      free(temp);
   }
}

void display(nodetype *top){
   printf("\n Elements in stack : ");
   while(top != NULL){
      printf(" %d ",top->info);
      top = top->next;
   }
}

void main(){

   nodetype *top = NULL;
   int val,ch,i = 1;
```

Stack Using Double Pointer

```
    while(i){
        printf("\nEnter your choice :\n1.PUSH \n2.POP \n3.DISPLAY \n4.EXIT\n");
        scanf("%d",&ch);
        switch(ch){
            case 1:
                printf("Enter the number to insert in new node :");
                scanf("%d",&val);
                push(&top,val);
                break;
            case 2:
                pop(&top);
                break;
            case 3:
                display(top);
                break;
            default : i=0;
        }
    }

}
```

OUTPUT :

```
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
1
Enter a number to insert in new node :11

Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
1
Enter a number to insert in new node :12

Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
```

Stack Using Double Pointer

1
Enter a number to insert in new node :13

Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
3

Elements in stack : 13 12 11
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
2

13 popped.

Enter your choice :
1.PUSH
2.POP
3.DISPLAY
4.EXIT
4