store
67

Create a struct in c take user input for USN, name, phone number

```
#include <stdio.h>
# include <string.h>
 struct student {
      char name[;10];
      char usN[;10];
      int ph ;
 } s1, s2, s3;
```

```
add()
{
  Sf(..)
  Pf(...)
}
```

```
Void Main()
{ printf ("enter your name");
 scanf ("%s", &s1.name);
                          add();
 Printf ("enter you USN");
 Scanf ("%s", &S1.USN);

 Printf ("enter your phonenumer");
 Scanf ("%d", &s1.ph);

  printf ("\n name is %ds", s1.name);
  printf ("\n USN is %ds", s1.USN);
  printf ("\n ph is %d", s1.ph);

  };
```

1/10/24

push

1. # include <stdio.h>; first include header statement.

2. # define MAX 3; define the constant MAX 3.

3. initialize your variables and define their type.
   int s [10], Top = -1, i, item, ch;

4. define function
   void push ( ) { }

5. print or display element to be displayed

6. Scanf or take user input.

7. incrementing the top
   top = top+1;

8. Allocating the value of the scanned item into the top
   s[top] = item;

```
if (top = max-1)
{ P ("stack overflow")
  return
}
Pf (enter element)
Sf (  )
top = top+1;
S[top] = itep;
}
```

pop

1. define function
   Void pop( ){ }

2. checking which item is on top of the stack
   item = s [top];

3. decrementing the top value
   top = top-1;

4. returning the poped element.
   return (item);

```
if (top = = -1)
   return (-1);
item = s[top];
top = top-1;
return (item);
```

store
67 1/10/24

check if stack is an overflow

1. void overflow () { define the function }.
2. checking if (top = max -1) checks if index = max idex a
3. print or display (stack is overflow)
4. return.

check if stack is an underflow

1. void underflow () { define the function }
2. checkin if stack is empty top = = -1
3. print or display (stack is underflow).
4. return

Display

1. void display () { define the function }
2. print or display (stack contents)
3. for (i = top ; i >= 0; i - -) print all entities ' in stack.
4. printf ("%.n\n", s [i]); prints all the entities
5. of the stack one below the other
6.

```
Push (item)
{   if ( top == MAX-1)
            prin("FULL");
    else
            Stack [++Top] = item.
}
```

```
if (top==-1) {
p (stack is empty)
return
 }.
pf (stack content)

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

pb ("%d\n", s[i]);

}
}
```

```
# include<stdio.h>
# include <con i o .h>
# include <stdlib.h>
int stack [10] , top = -1, i , item ;
# define max 3

 void push () {
      if (top== max-1) {
           printf ("stack overflow \n");
        }
      else {
        top++;
        printf ("enter element to push : ");
        scanf ("%d ", & item);
        Stack[top] = item;
        }
    }
int pop () {
     if (top ==-1) {
         printf (" stack underflow \n");
         return -1;
     }
       item = stack [top] ;
       top = top-1;
       return (item );
  }
```

store
67

```
Void display () {
    if (top== -1){
        printf (" stack if empty\n );
    }
    else  {
        printf (" the stack is : \n");
         for ( i = top ; i > -1; ~~top~~ i - - ) {
            printf ("%d \n", stack [i] );
         }
       }
    }

 Void main () {
 while (1)  {
     int userInput;
     printf("enter \n1 to push, \n2 to pop, \n3 to
            display, and \n4 to exit \n");

     Scanf ("%d ", &userInput );
     Switch (userinput) {
       case1 : push ();
             break;
       case2 : item = pop();
             if (item!= -1) {
                printf ("the popped element is %d \n", item);
             }
           break;
       case 3: display() ;
             break ;
       case 4: exit (0) ;
             break;
     }
  }
}
```

Output,

```
Enter
1 to push ,
2 to pop ,
3 to display , and
4 to exit

 1
Enter Element to push: 11

Enter
1 to push ,
2 to pop ,
3 to display , and
4 to exit.
1
Enter Element to push: 22.

Enter
1 to push ,
2 to pop ,
3 to display , and
4 to exit
1
Enter Element to push: 33
```

```
Enter
1 to push,
2 to pop,
3 to display, and
4 to exit,
 1
stack overflow

Enter
1 to push,
2 to pop,
3 to display, and
4 to exit

3

33
22
11

Enter
1 to push
2 to pop
3 to display and
4 to exit
2
The popped element is 33

Enter
1 to push
2 to pop
3 to display and
4 to exit
2
popped element is 22
```

```
Enter
1 to push,
2 to pop,
3 to display, and
4 to exit

2
the popped element is 11

Enter
1 to push,
2 to pop,
3 to display, and
4 to exit
2
Stack underflow.
```

01/10/2025