

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

b)

#include <stdio.h>

#include <string.h>

struct student {

char name[10];

char usn[10];

int ph;

} s1, s2, s3;

add()

{

if (...

printf(...)

}

void main()

{ printf("enter your name");

scanf("%s", &s1.name);

add();

printf("enter your usn");

scanf("%s", &s1.usn);

printf("enter your phonenumber");

scanf("%d", &s1.ph);

~~printf("enter your name");~~

printf("\n name is %s", s1.name);

printf("\n usn is %s", 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 if (top == max-1)
6. scanf or take user input - { p ("stack overflow");
7. incrementing the top
top = top + 1; return
8. Allocating the value of the scanned item
into the top
s[top] = item; }
p (enter element)
scanf ()
top = top + 1;
s[top] = item;

pop

1. define function
void pop() { }
 2. checking which item is on top of the stack
item = s[top];
 3. ~~decrementing~~ incrementing the top value
top = top - 1;
 4. returning the popped element -
return (item);
- if (top == -1)
return (-1);
item = s[top];
top = top - 1;
return (item);

check if stack is an overflow

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

check if stack is an underflow

1. void underflow() { define the function }
2. check 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 (">d\n", s[i]); prints all the entities
5. of the stack one below the other.
- 6.

```

push(item)
{
    if (top == max-1)
        printf("Full");
    else
        stack[++top] = item;
}
    
```

```

if (top == -1) {
    printf("Stack is empty");
    return;
}
    
```

```

printf("Stack content")
    
```

```

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

```

printf(">d\n", s[i]);
    
```

```

}
}
    
```

```
#include <stdio.h>
#include <conio.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);
}
```


3

3 3 3

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

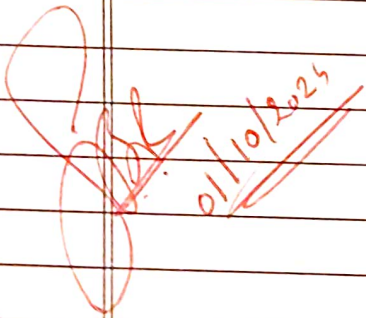
the popped element is 11

Enter

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

2

Stack underflow.

 01/10/2024