

week 2

① Swapping using pointers

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

```
void Swap (int *x, int *y) {
```

```
    int t;
```

```
    int * t = * x;
```

```
    *x = *y;
```

```
    *y = t;
```

```
}
```

```
int main () {
```

```
    int *x, *y;
```

```
    printf ("Enter two numbers that has to be swapped: ");
```

```
    scanf ("%d %d", &x, &y);
```

```
    printf ("x = %d and y = %d", x, y);
```

```
    Swap (*x, *y);
```

```
    printf ("After swapping x = %d and y = %d", x, y);
```

```
    return 0;
```

```
}
```

Output

Enter two numbers that has to be swapped: 1
2

x = 1 and y = 2

After swapping x = 2 and y = 1

(2) Malloc()

$$pt = (\text{int}^*) \text{malloc}(n * \text{sizeof}(\text{int}));$$
calloc()

$$pt = (\text{int}^*) \text{calloc}(n, \text{sizeof}(\text{int}));$$
free()

$$\text{free}(pt);$$
realloc()

$$pt = (\text{int}^*) \text{malloc}(...);$$
~~$$pt = (\text{int}^*) \text{realloc}(...);$$~~
~~$$pt = (\text{int}^*) \text{realloc}(pt, n * \text{sizeof}(\text{int}));$$~~
~~#include <stdio.h>~~

```
int Main() {
    int *pt;
    int i, n, m;
    printf("Enter no of elements:");
    scanf("%d", &n);
    pt = (int*) malloc(n * sizeof(int));
    if (pt == NULL) {
        printf("Memory not allocated by malloc");
    }
}
```

~~else {~~
~~for (i=0; i<n; i++)~~
~~printf("Enter elements:");
scanf("%d", &pt[i]);~~
~~}~~
~~printf("Memory successfully allocated using malloc");~~
~~printf("The no of elements are:");~~
~~for (i=0; i<n; i++) {~~

```
printf ("%d", pti[i]);  
}  
free(pti);  
printf (" memory freed");  
pti=(int *)calloc(n, sizeof(int));  
for(i=0; i<n; i++)  
{
```

```
printf ("%d", pti)  
printf (" n entri elements %d: ", n);  
scanf ("%d", &pti[i]);
```

}

```
printf ("n Successful allocated using calloc ");  
printf ("n entri no of elements for realloc: ");  
scanf ("%d", &m);  
pti = (int *)realloc(pti, m * sizeof(int));  
printf ("n successful reallocated memory ");  
printf ("n n entri more elements: ");  
for(i=n; i<m; i++)  
{
```

{

```
printf ("n entri elements: ");  
scanf ("%d", &pti[i]);  
}
```

```
printf (" elements are: ");  
for(i=0; i<m; i++)  
{
```

```
printf ("%d", pti[i]);  
}
```

}

```
return 0;  
}
```

output

Enter no of elements = 3

Enter elements = 1

2

3

Successfull allocated using malloc

The no of elements are : 1, 2, 3, memory freed

Enter elements 3 : 1

2

3

Successfull allocated using calloc

Enter no of elements for realloc = 5

Successfull reallocated memory

Enter more elements:

Enter elements : 4

5

The no of elements are : 1, 2, 3, 4, 5,

③ Stack operations push, pop, display.

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

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

```
#define size 10
```

```
void push();
```

```
void pop();
```

```
void display();
```

```
int main()
```

```
{
```

```
int top = -1, st[size];
```

```
int choice;
```

while()

{

```
printf("In perform operations on Stack = ");  
printf("In 1.push m2, pop in 3. display\n4. EXIT ");  
printf("In In enter choice = ");  
scanf("%d", &choice);
```

switch (choice)

{

case 1:

```
push();
```

```
break;
```

case 2:

```
pop();
```

```
break;
```

case 3:

```
display();
```

```
break;
```

case 4:

```
exit(0);
```

default:

```
printf("In invalid choice !!");
```

}

}

return 0; }

void push()

{

```
int x;
```

```
if (top == size - 1)
```

{

```
printf("\noverflow!!");
```

}

else

{

printf ("\n enter the element to be added onto stack : ");

scanf ("%d", &x);

top = top + 1;

st [top] = x;

}

y

void pop()

{ if (top == -1)

{

printf ("\n underflow ! ");

}

else

{

printf ("\n popped element : %d ", st [top]);

top = top - 1;

}

B

void display()

{

if (top == -1)

{

printf ("\n underflow ! ");

}

else

{

printf ("\n elements present in stack : \n ");

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

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

}

B

Output

1. push
2. pop
3. display
4. end

enter choice = 1

enter element to be added onto stack = 5

1. Push
2. pop
3. display
4. end

enter choice = 1

enter element to be added onto stack = 4

1. push
2. pop
3. display
4. end

enter choice = 2

popped element = 4

1. push
2. pop
3. display
4. end

enter choice = 3

element present in stack = 5

1. push
2. pop
3. display
4. end

Enter choice = 4.

~~Sneha~~
21/12/23