

LAB-1:-

Date ___/___/___
Page _____

① Output:-

Enter 1: acc creation

2: Deposit

3: withdrawal

4: bal Enq

5: Exit

Enter account number: 5678

Enter balance: 10,000

Enter 1: acc creation

2: Deposit

3: withdrawal

4: bal Enq

5: Exit

2.

Enter account number: 5678

Enter amt to be deposited: 1000

Enter 1: acc creation

2: Deposit

3: withdrawal

4: bal Enq

5: Exit

② Output:-

Enter the number of strings: 5

Enter 5 strings:

hello java world javascript html.

sorted strings lexicographically

hello .html java javascript world.

③ Output :-

Enter order: 2 2

Enter elements : 12 3 4

Enter key : 6

Key not found

④ Enter the array :

1 2 3

4 5 6

7 8 9

Enter the element to search for : 2

Enter 2 is present in 2D array

⑤ Output :-

Enter no : 6

Enter the array elements

10 7 5 2 8 4

Enter element to search for:

5

Enter 5 found at index 2.

⑥ Output :-

Enter the no of element in array : 10

Enter the elements : 1 10 2 6 7 9 20 2 2 4

Enter no to be searched : 7

7 is present at index 5

The search 7 is present 1 time

⑦ Output:-

Enter the value of n: 6

Enter the element:

1 3 5 7 9 11

Enter the elements to be searched (key):

5

5 is present at index 1

⑧ Output:-

Enter size of array: 5

Enter elements: 12 350 -1

minimum: -1

maximum: 350

Date / /
Page

② Dynamic

```
#include <cs
void main
void Call
void Rea
void main
{
    int *
    int n
    printf
    scanf
    Make
    Call
    Rea
}
void
{
```

Enter the values of a and b:

9

6

The swap values are: 69

3

3

for
for
for

② Dynamic allocation :-

```
#include <stdio.h>
void Malloc (int);
void Calloc (int);
void Realloc (int);
void main ()
```

```
{
    int *ptr;
    int n, i;
    printf ("enter the value of n: ");
    scanf ("%d", &n);
    Malloc (n);
    Calloc (n);
    Realloc (n);
}
```

```
void Malloc (int n)
```

```
{
    int *ptr;
    int i;
    int arr[n];
    ptr = (int*) malloc (n * sizeof (int));
    for (i = 0; i < n; i++)
```

```
{
    ptr[i] = i + 1;
}
```

```
printf ("Malloc");
printf ("the elements of array are ");
for (i = 0; i < n; i++) {
    printf ("%d ", ptr[i]);
}
```

```
printf ("\n");
}
```

```

void calloc (int n)
{
    int * ptr;
    int i;
    int arr [n];
    ptr = (int *) calloc (n, size of (int));
    for (i=0; i<n; i++)
    {
        ptr[i] = i+1;
    }
    printf("\n");
    free (ptr);
}

```

```

void Realloc (int n)
{
    int * ptr;
    int i;
    int arr [n];
    n = 10;
    ptr = (int *) realloc (ptr, n * size of (int));
    for (i=5; i<n; i++)
    {
        ptr[i] = i+1;
    }
    ptr = ptr;
    for (i=0; i<n; i++)
    {
        printf("%d", ptr[i]);
    }
    free (ptr);
}

```


Output:-

Enter the value of n:

5

malloc

the elements of array are:

1 2 3 4 5

calloc

the elements of array are:

1 2 3 4 5 0

realloc

the elements of array are:

1 2 3 4 5 6 7 8 9 10

③ Stack

#include <stdio.h>

#include <stdlib.h>

#define SIZE 4

int top = -1;

int arr[SIZE];

void push();

void pop();

void show();

void main()

{

int ch;

while(1)

{

printf("operation on the stack: \n");

printf("1. push the element \n 2. pop the element \n 3.

show \n 4. End \n");

NP
21/12/2023

```
printf("Enter the choice: \n");
scanf("%d", &ch);
```

Switch (ch)

```
{
```

Case 1:

```
push();
break;
```

Case 2:

```
pop();
break;
```

Case 3:

```
show();
break;
```

Case 4:

```
exit(0);
```

default:

```
printf("Invalid choice \n");
```

```
}
```

```
}
```

```
}
```

```
void push()
```

```
{
```

```
int x;
```

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

```
{
```

```
printf("Overflow \n");
```

```
}
```

else

```
{
```

```
printf("Enter the element to be added: \n");
```

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



```
top = top + 1;  
inp_array[top] = x;
```

```
}
```

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

```
{  
    printf("Underflow\n");
```

```
}
```

```
else
```

```
{
```

```
    printf("popped element: %d\n", inp_array[top]);  
    top = top - 1;
```

```
}
```

```
}
```

```
void show()
```

```
{
```

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

```
{
```

```
        printf("Underflow\n");
```

```
}
```

```
else
```

```
{
```

```
    printf("Elements in the stack are: \n");
```

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

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

```
}
```

```
}
```

```
ed: \n");
```

Output :-

Enter the choice :

1

Enter the element to be added :

5

Operations on the stack :

1. push

2. Pop

3. Show

4. End

Enter the choice :

1

Enter the element to be added :

6

Operations on Stack :

1. push

2. pop

3. Show

4. end

Enter the choice :

3

elements in stack are :

6

5