

Program 1:
write a program to swap two numbers using C

(21/12/23)

```
#include <stdio.h>
void swap(int *, int *);
void main()
{
    int a, b;
    printf("Enter two numbers:\n");
    scanf("%d %d", &a, &b);
    printf("Before swapping, values of a and b are: %d %d", a, b);
    swap(&a, &b);

}
void swap(int *p, int *q)
{
    int temp;
    temp = *p;
    *p = *q;
    *q = temp;
    printf("After swapping, values of a and b are: %d %d", *p, *q);
}
```

Output:

Enter two numbers:

4 5

Before swapping, values of a and b are: 4 5

After swapping, values of a and b are: 5 4

(21/12/23)

Program 2:

write a program for dynamic memory allocation

```
#include <stdio.h>
void malloc(int);
void calloc(int);
//void realloc();
void main()
{
    int n;
    printf("Enter the value of n:\n");
}
```

```
scanf("%d", &n);
mallocEx(n);
callocEx(n);
}
void MallocEx(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 dynamic memory allocation (%d)",

printf("The elements of the array are : (%d)",

```
for(i=0; i<n; i++)
```

```
{ printf("%d", ptr[i]);
```

```
printf("\n");
free(ptr);
```

```
}
```

```
void callocEx(int n)
```

```
{
```

```
int *ptr;
```

```
int i;
```

```
int arr[n];
```

```
ptr = (int *)calloc(n, sizeof(int));
```

```
for(i=0; i<n; i++)
```

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

```
}
```

printf("calloc dynamic memory allocation (%d)",

printf("The elements of the array are : (%d)",

```
for(i=0; i<n; i++)
```

```
{ printf("%d", ptr[i]);
```

```
}
```

```

printf("1n");
printf("Realloc dynamic memory allocation(n");
printf("the elements of the array are: \n");
n=10;
ptr=(int*)realloc(ptr, n+sizeof(int));
for(r=5; i<n; i++)
{
    ptr[i] = i+1;
}
for(r=0; i<n; i++)
{
    printf("%d", ptr[i]);
}
free(ptr);
}

```

Output:

Enter the value of n:

5
malloc dynamic memory allocation
The elements of the array are:

1 2 3 4 5

Callc dynamic memory allocation
The elements of the array are:

1 2 3 4 5

Realloc dynamic memory allocation
The elements of the array are:

1 2 3 4 5 6 7 8 9 10

((or/9) realloc(")) (free: free)

Program 3:
write a program for stack implementation

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 4
int top = -1;
int inp_array[SIZE];
void push();
void pop();
void show();

void main()
{
    int ch;
    while (1)
    {
        printf("Operations on the stack:\n");
        printf("1. Push the element\n 2. Pop the element\n 3. Show\n 4. End\n");
        printf("Enter the choice: ");
        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;
        inf_array[top] = x;
    }
}

void pop()
{
    if (top == -1)
    {
        printf("Underflow\n");
    }
    else
    {
        printf("Popped element: %d\n", inf_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", inf_array[i]);
    }
}

```

Output:

Operations on the stack:

1. Push the element
2. Pop the element
3. Show
4. End

Enter the choice:

1 Enter the element to be added in the stack : ("alpha")

4

Operations on the stack:

1. Push the element
2. Pop the element
3. Show

4. End

Enter the choice:

1 Enter the element to be added in the stack : ("alpha")

5

Operations on the stack:

1. Push the element
2. Pop the element
3. Show
4. End

Enter the choice

2

Popped element : 5

Operations on the stack

1. Push the element
2. Pop the element
3. Show
4. End

Enter the choice :

3

Elements in the stack are :

4

Operations on the stack:

1. Push the element
2. Pop the element
3. Show

H. End

Enter the choice:

H

N
29/12/2023