21/12/23

1. Swapping using pointers

```
#include <stdio.h>
void swapreference (int *a, int *b);
void main()
{
    int a=10;
    int b=20;
    printf("values of a and b before swapping are %d and %d\n",a,b);
    swapreference (&a, &b);
    printf("values of a and b after swapping are %d and %d \n", a, b);
}

void swapreference (int *a, int *b)
{
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
    printf("values of a and b inside function are %d and %d\n",*a,*b);
}
```

Output

values of a and b before swapping are 10 and 20
values of a and b inside ~~call by~~ function are 20 and 10
values of a and b after swapping ~~by call~~ are 20 and 10

2.
```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    int *p, *q;
```

```
int n;
int i;
printf("read n:");
scanf("%d",&n);
p=(int*) malloc (n* sizeof(int));
printf("enter %d elements:", n);
for(i=0; i<n; i++)
    scanf("%d", p+i);
q=(int*) calloc (n, sizeof(int));
printf("enter %d elements:", n);
for(i=0; i<n; i++)
    scanf("%d", q+i);
realloc(p, 7* sizeof(int));
printf("enter 7 elements:");
for(i=0; i<7; i++)
    scanf("%d", p+i);
free(p);
free(q);
}
```

Output

```
read n: 5
enter 5 elements: 1 2 3 4 4
enter 5 elements: 1 2 6 5 8
enter 7 elements: 1 3 2 4 5 6 6
```

3.
```
#include <stdio.h>
#define max 5
int top = -1;
int s[max];
void push (int value)
{
  if (top == max-1)
      printf("stack overflow cant push");
  else {
      top = top+1;
      s[top] = value;
  }
}
void pop()
{
    int value;
    if (top == -1)
        printf("stack is underflow cant pop\n");
    else
    {
       value = s[top];
       top = top-1;
       printf("\n %d is popped\n", value);
    }
}
void isempty()
{
    if (top == -1)
        printf("stack is empty\n");
}
void isfull()
{
    if (top == max-1)
        printf("stack is full\n"); }
```

```
void display()
{
    if (top == -1)
        printf("stack is underflow\n");
    else
    {
        printf("\n stack elements are: ");
        for(int i=0; i<=top; i++)
            printf("%d\t", s[i]);
    }
}
void main()
{
    int ~~xno~~;
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    printf("enter a no: ");
    scanf("%d", &no);
    push(no);
    display();
```

```
pop();
pop();
pop();
pop();
isempty();
isfull();
display();
pop();
pop();
}
```

Output

enter a no: 10
enter a no: 20
enter a no: 30
enter a no: 40
enter a no: 50
enter a no: 60
stack is overflow cant push
stack elements are: 10 20 30 40 50
50 is popped
40 is popped
30 is popped
20 is popped.
stack elements are: 10 10 is popped.
stack is underflow. cant pop

21/12/23