**LABORATOR 3 – Tema (Flavia Frigura)**

#include <stdio.h>  
#include <stdint.h>  
#include <ctype.h>  
#include <string.h>  
#include <stdlib.h>  
#include <math.h>  
  
**APLICATIA 3.1**

#define MAX\_SIZE 100  
  
int top=-1;  
int stack[MAX\_SIZE];  
  
void push(int item)  
{  
    if(top==MAX\_SIZE-1)  
    {  
        printf("Stiva este plina\n");  
        return;  
    }  
    stack[++top]=item;  
}  
  
int pop()  
{  
    if(top==-1)  
    {  
        printf("Stiva este goala\n");  
        return -1;  
    }  
    return stack[top--];  
}  
  
void display()  
{  
    if(top==-1)  
    {  
        printf("Stiva este goala\n");  
        return;  
    }  
    printf("Elementele din stiva sunt: ");  
    for(int i=top;i>=0;i--)  
        printf("%d ",stack[i]);  
    printf("\n");  
}  
  
  
  
int main()  
{  
    push(10);  
    push(20);  
    push(30);  
    display();  
    printf("Elementul extras din stiva: %d\n",pop());  
    display();  
    return 0;  
}

**APLICATIA 3.2**

#define MAX\_SIZE 100  
  
int front=-1,rear=-1;  
int queue[MAX\_SIZE];  
  
void enqueue(int item)  
{  
    if(rear==MAX\_SIZE-1)  
    {  
        printf("Coada este plina\n");  
        return;  
    }  
    if(front==-1) front=0;  
    queue[++rear]=item;  
}  
  
int dequeue()  
{  
    if(front==-1||front>rear)  
    {  
        printf("Coada este goala\n");  
        return -1;  
    }  
    return queue[front++];  
}  
  
void display()  
{  
    if(front==-1||front>rear)  
    {  
        printf("Coada este goala\n");  
        return;  
    }  
    printf("Elementele din coada sunt: ");  
    for(int i=front;i<=rear;i++)  
        printf("%d ",queue[i]);  
    printf("\n");  
}  
  
int main()  
{  
    enqueue(10);  
    enqueue(20);  
    enqueue(30);  
    display();  
    printf("Elementul extras din coada: %d\n",dequeue());  
    display();  
    return 0;  
}

**APLICATIA 3.3**  
  
#define MAX\_SIZE 100  
  
typedef struct  
{  
    char s[100];  
    int rand;  
}caractere;  
  
int top=-1;  
caractere stack[MAX\_SIZE];  
  
void push(char item[])  
{  
    if(top==MAX\_SIZE-1)  
    {  
        printf("Stiva este plina\n");  
        return;  
    }  
    top++;  
    strcpy(stack[top].s,item);  
    stack[top].rand=top;  
}  
  
void pop()  
{  
    if (top == -1)  
    {  
        printf("Stiva este goala\n");  
        return;  
    }  
    printf("Undo: s-a eliminat %s\n", stack[top].s);  
    top--;  
}  
  
  
void display()  
{  
    if(top==-1)  
    {  
        printf("Stiva este goala\n");  
        return;  
    }  
    printf("Elementele din stiva sunt: \n");  
    for(int i=top;i>=0;i--)  
        printf("%s\n",stack[i].s);  
    printf("\n");  
}  
  
  
int main()  
{  
    push("Acesta este primul rand");  
    push("Acesta este al doilea");  
    display();  
    pop();  
    display();  
    push("Am recorectat randul 2");  
    push("Randul 3 e aici");  
    display();  
    pop();  
    display();  
    push("Acesta este al treilea rand");  
    display();  
    return 0;  
}

**APLICATIA 3.4**  
  
#define MAX\_SIZE 100  
  
typedef struct  
{  
    int id;  
    char mesaj[256];  
}informatie;  
  
int front=-1,rear=-1;  
informatie queue[MAX\_SIZE];  
  
void enqueue(char item[])  
{  
    if(rear==MAX\_SIZE-1)  
    {  
        printf("Coada este plina\n");  
        return;  
    }  
    if(front==-1) front=0;  
    rear++;  
    strcpy(queue[rear].mesaj,item);  
}  
  
void dequeue()  
{  
    if(front==-1||front>rear)  
    {  
        printf("Coada este goala\n");  
        return;  
    }  
    printf("S-a transmis receptorul si s-a eliminat %s\n", queue[front].mesaj);  
    front++;  
}  
  
void display()  
{  
    if(front==-1||front>rear)  
    {  
        printf("Bufferul este gol\n");  
        return;  
    }  
    printf("Elementele din buffer sunt: ");  
    for(int i=front;i<=rear;i++)  
        printf("%s\n",queue[i].mesaj);  
    printf("\n");  
}  
  
int main()  
{  
    enqueue("Prima linie");  
    enqueue("A doua");  
    enqueue("A treia");  
    display();  
    dequeue();  
    display();  
    return 0;  
}