|  |  |
| --- | --- |
|  | |
| Assignment No: 01     1. A program of ‘Stack’ using array. 2. A program of ‘Queue’ using array. | |
| Date of Performance: 27/01/2019  Date of Submission: 03/02/2019 | Student ID: 17-02-04-058  Group: B1 |

**No.1:**

#include <stdio.h>

int MAXSIZE = 5;

int stack[5];

int top = -1;

int isempty()

{

if(top == -1)

return 1;

else

return 0;

}

int isfull()

{

if(top == MAXSIZE)

return 1;

else

return 0;

}

int peak()

{

return stack[top];

}

int pop()

{

int data;

if(!isempty())

{

data = stack[top];

top = top - 1;

return data;

}

else

{

printf("Could not retrieve data, Stack is empty.\n");

}

}

int push(int data)

{

printf("Enter item to be inserted: ");

scanf("%d",&data);

if(!isfull())

{

top = top + 1;

stack[top] = data;

}

else

{

printf("Could not insert data, Stack is full.\n");

}

}

int elements(int data)

{

{

printf("\nElements: \n");

while(!isempty())

{

int data= pop();

printf("%d\n",data);

}

}

}

int main()

{

int ch;

int loop;

loop=1;

int data;

do

{

printf("\n \*\*\*STACK OPERATIONS");

printf("\n 1. PUSH");

printf("\n 2. POP");

printf("\n 3. PEAK");

printf("\n 4. Elements");

printf("\n 5. EXIT");

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n Enter your choice: ");

scanf("%d", &ch);

switch (ch)

{

case 1:

push(data);

break;

case 2:

pop();

break;

case 3:

printf("Element at the top: %d",peak());

break;

case 4:

elements(data);

break;

case 5:

printf("THANK YOU");

loop=0;

return;

default:

printf("Invalid choice");

}

}

while(loop);

return 0;

}

**No.2:**

#include <stdio.h>

#define SIZE 5

int items[SIZE];

int front = -1, rear =-1;

int isFull()

{

if( (front == rear + 1) || (front == 0 && rear == SIZE-1))

return 1;

else

return 0;

}

int isEmpty()

{

if(front == -1)

return 1;

else

return 0;

}

void enQueue(int element)

{

printf("\nEnter the element\n");

scanf("%d",&element);

if(isFull())

printf("\n Queue is full!! \n");

else

{

if(front == -1)

front = 0;

rear = (rear + 1) % SIZE;

items[rear] = element;

printf("\n Inserted -> %d", element);

}

}

int deQueue()

{

int element;

if(isEmpty())

{

printf("\n Queue is empty !! \n");

return(-1);

}

else

{

element = items[front];

if (front == rear)

{

front = -1;

rear = -1;

}

else

{

front = (front + 1) % SIZE;

}

printf("\n Deleted element -> %d \n", element);

return(element);

}

}

void display()

{

int i;

if(isEmpty())

printf(" \n Empty Queue\n");

else

{

printf("\n Front -> %d ",front);

printf("\n Items -> ");

for( i = front; i!=rear; i=(i+1)%SIZE)

{

printf("%d ",items[i]);

}

printf("%d ",items[i]);

printf("\n Rear -> %d \n",rear);

}

}

int main()

{

int choice,element;

while(choice != 4)

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1.insert an element\n2.Delete an element\n3.Display the queue\n4.Exit\n");

printf("\nEnter your choice ?");

scanf("%d",&choice);

switch(choice)

{

case 1:

enQueue(element);

break;

case 2:

deQueue();

break;

case 3:

display();

break;

case 4:

printf("THANK YOU");

return;

break;

default:

printf("\nEnter valid choice??\n");

}

}

}