**PROGRAM:**

**Array implementation of stacks**

**Input:**

#include<stdio.h>

#define max 4

int stack[50];top=-1;

int push();

int pop();

int stack\_overflow();

int stack\_underflow();

int peek();

int display();

int main()

{

int item,choice=0;

printf("\n1.push\n");

printf("2.pop\n");

printf("3.stack overflow\n");

printf("4.stack underflow\n");

printf("5.peek\n");

printf("6.display\n");

printf("7.quit\n");

do

{

printf("\nenter your choice:");

scanf("%d",&choice);

printf("\n");

switch(choice)

{

case 1: push();

break;

case 2: pop();

break;

case 3: stack\_overflow();

break;

case 4: stack\_underflow();

break;

case 5: peek();

break;

case 6: display();

Expt no: 2

break;

case 7: exit(0);

break;

default:printf("invalid input\n");

}

}while(choice!=7);

return 0;

}

int push(int item)

{

printf("enter the number to be inserted:");

scanf("%d",&item);

printf("\n");

if(top==max-1)

{

printf("stack overflow\n");

return;

}

else

top=top+1;

stack[top]=item;

}

int pop(int item)

{

if(top==-1)

{

printf("stack underflow\n");

return;

}

else

stack[top]=item;

top=top-1;

printf("element deleted\n");

return item;

}

int stack\_overflow()

{

if(top==max-1)

printf("stack overflow\n");

else

printf("stack not overflow\n");

return;

}

int stack\_underflow()

{

if(top==-1)

printf("stack underflow\n");

else

printf("stack not underflow\n");

return;

}

int peek()

{

if(top==-1)

printf("stack underflow\n");

else

printf("topmost element=%d",stack[top]);

}

int display()

{

int i;

if(top==-1)

printf("stack underflow\n");

else

{

printf("stack:");

for(i=0;i<=top;i++)

{

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

}printf("\n");

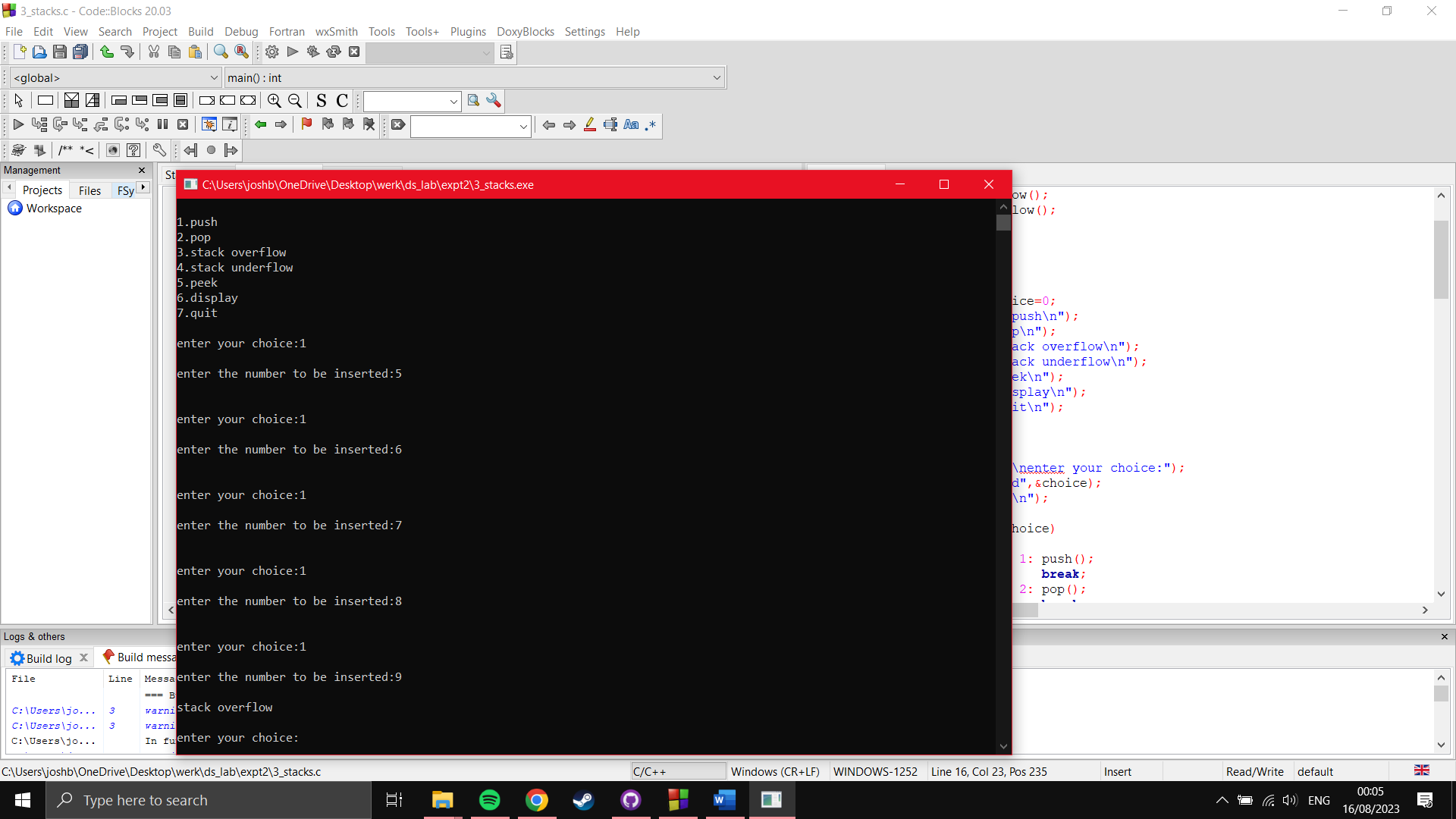
}

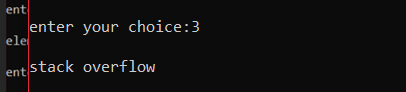
}

Expt no: 2

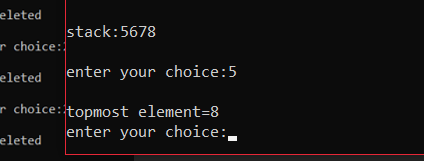
**Output:**

Push:

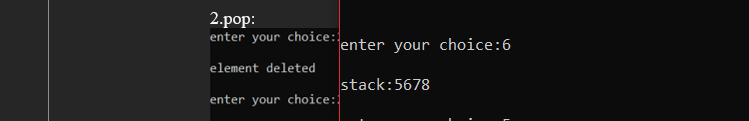


Stack overflow:

Peek:

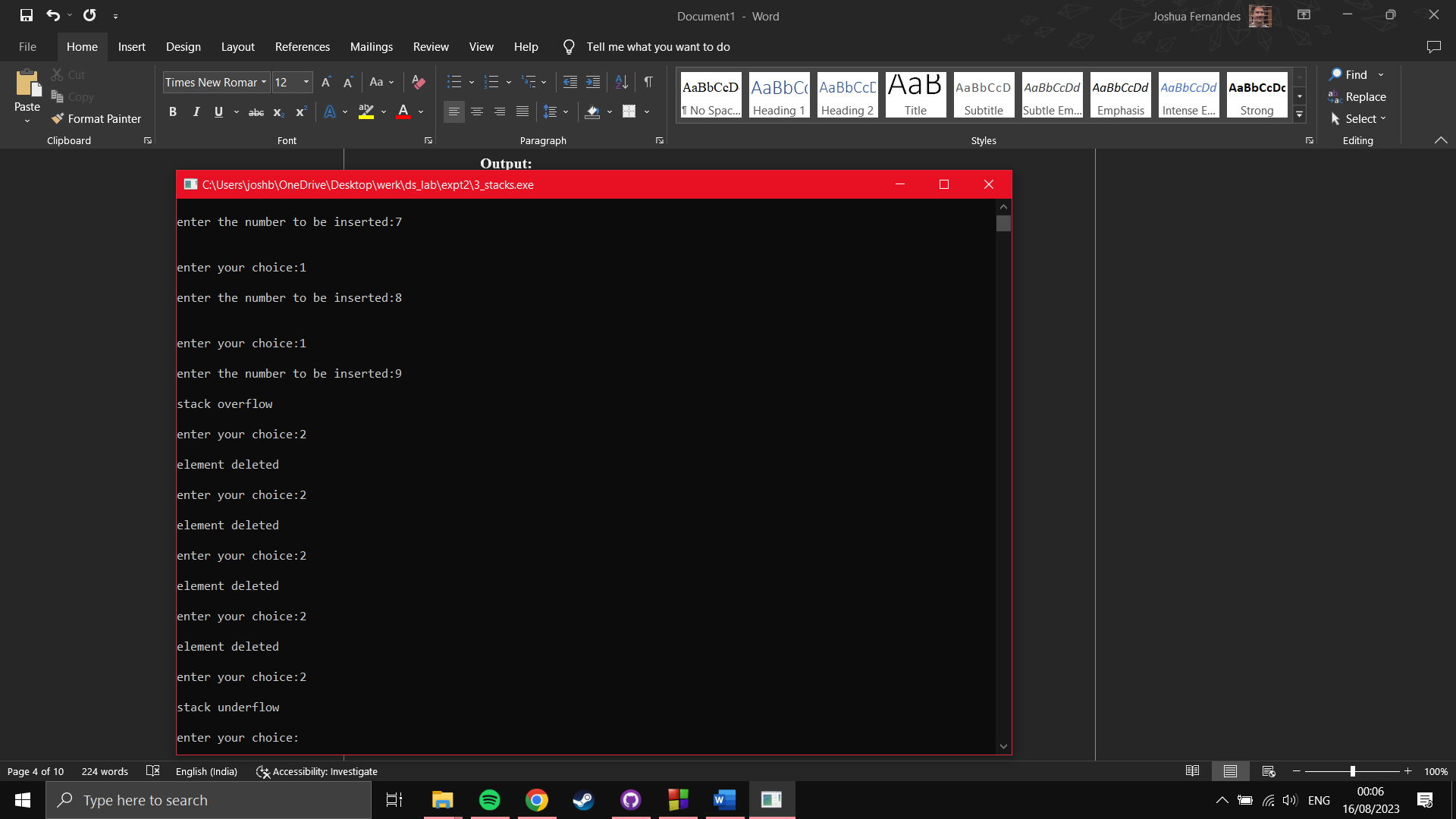


Display:

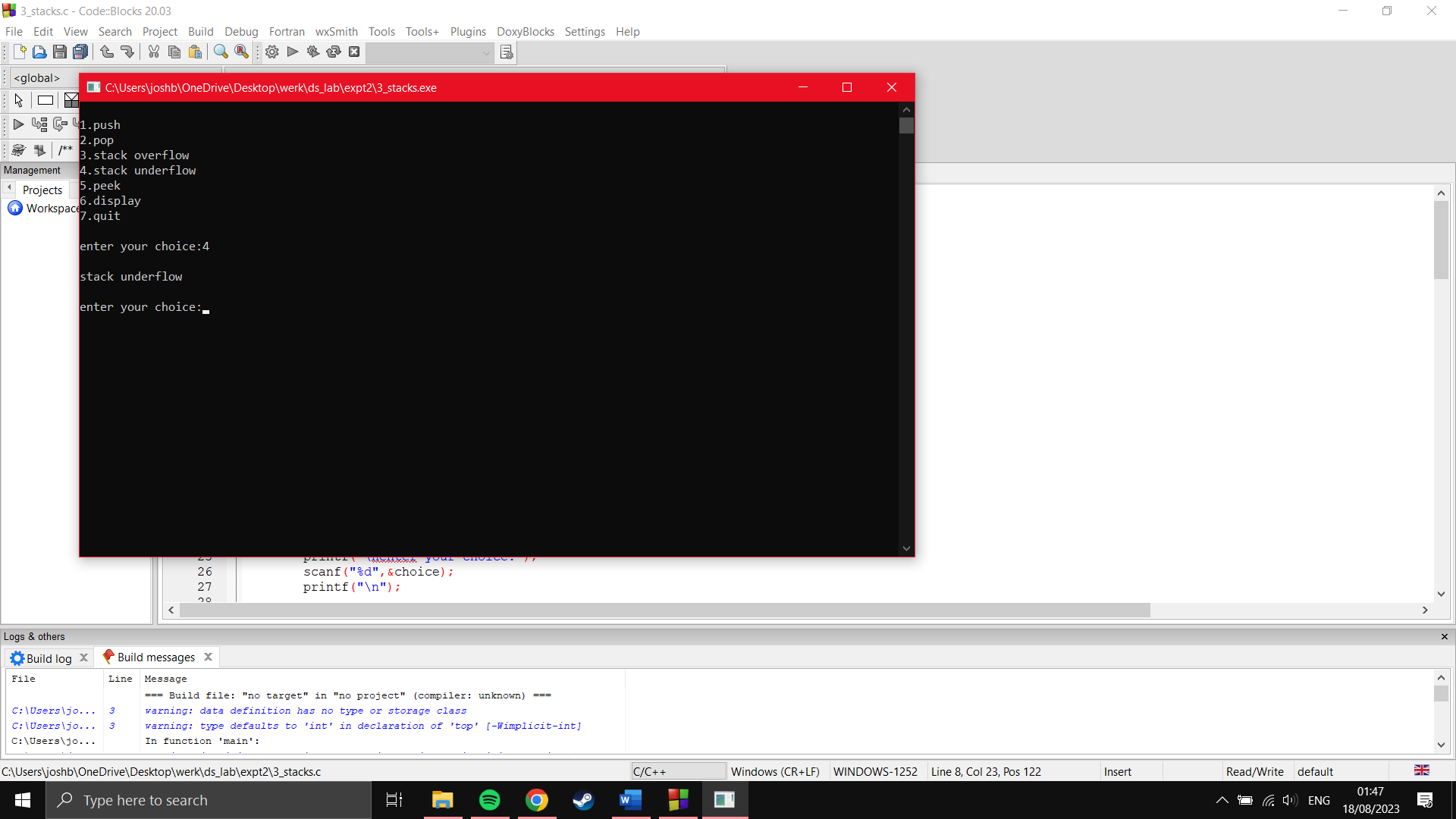


Pop:

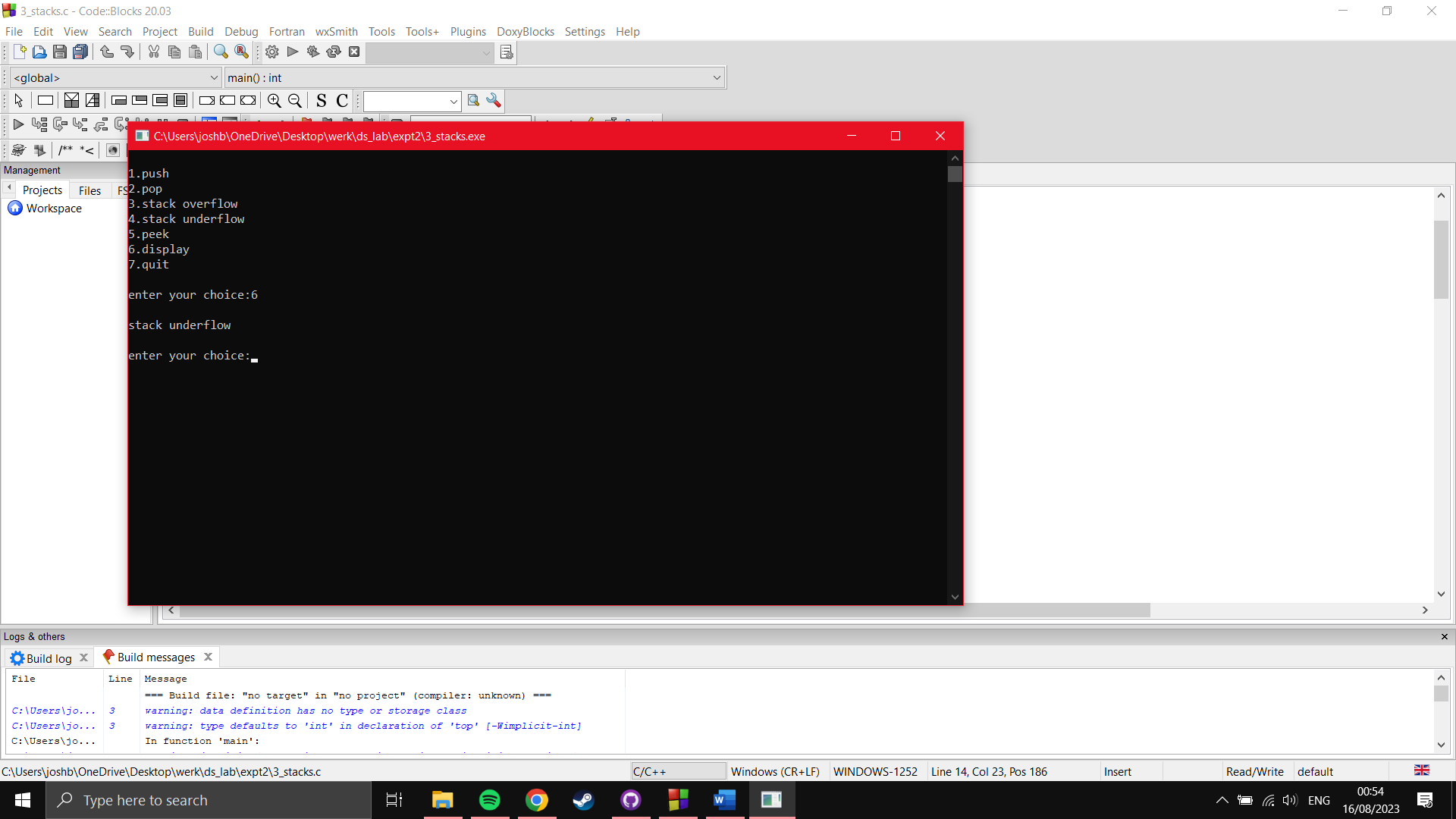
Expt no: 2



Stack underflow:



Display(for empty stack):



**PROGRAM:**

**Array implementation of queues**

**Input:**

#include<stdio.h>

#define max 4

int queue[100],front=-1,rear=-1;

int enqueue();

int dequeue();

int is\_full();

int is\_empty();

int peek();

int display();

int main()

{

int item,choice;

printf("\n1.enqueue\n");

printf("2.dequeue\n");

printf("3.queue overflow\n");

printf("4.queue underflow\n");

printf("5.peek\n");

printf("6.display\n");

printf("7.quit\n");

do

{

printf("\nenter your choice:");

scanf("%d",&choice);

printf("\n");

switch(choice)

{

case 1: enqueue();

break;

case 2: dequeue();

break;

case 3: is\_full();

break;

case 4: is\_empty();

break;

case 5: peek();

Expt no: 2

break;

case 6: display();

break;

case 7: exit(0);

break;

default:printf("invalid input\n");

}

}while(choice!=7);

return 0;

}

int enqueue(int item)

{

if(rear==max-1)

printf("queue overflow\n");

else if (front==-1)

{

front=0;

printf("queue was previously empty");

}

else

{

printf("enter the number to be inserted:");

scanf("%d",&item);

rear++;

queue[rear]=item;

}

}

int dequeue(int item)

{

if(front==-1||front>rear)

printf("queue underflow\n");

else

printf("the deleted element is:%d\n",queue[front]);

front++;

}

int is\_full(int item)

{

Expt no: 2

if(rear==max-1)

printf("queue overflow\n");

else

printf("queue not overflow\n");

}

int is\_empty(int item)

{

if(front==-1||front>rear)

printf("queue underflow\n");

else

printf("queue not underflow\n");

}

int peek(int item)

{

if(front==-1||front>rear)

printf("queue underflow\n");

else

printf("the front element=%d\n",queue[front]);

}

int display()

{

int i;

if (front==-1||front>rear)

printf("queue underflow\n");

else

{

printf("queue:");

for(i=front;i<=rear;i++)

{

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

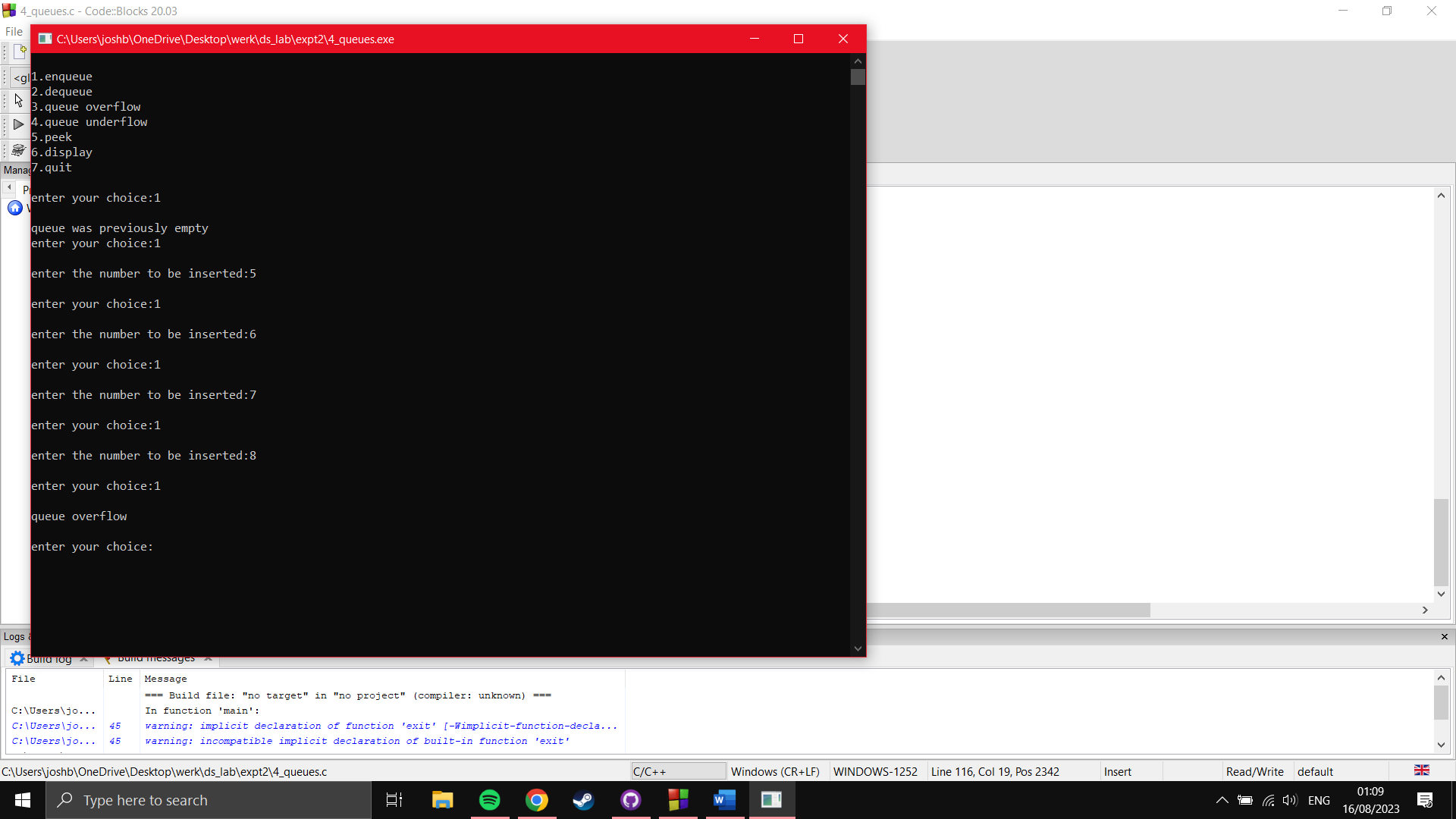
}

}printf("\n");

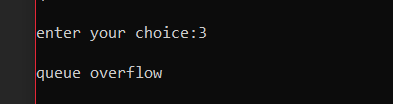
}

**Output:**

Enqueue:



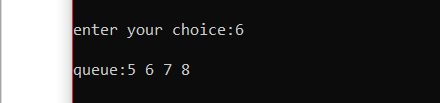
Queue overflow:



Peek:

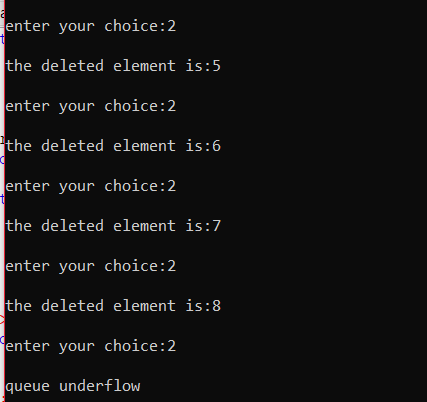


Display:

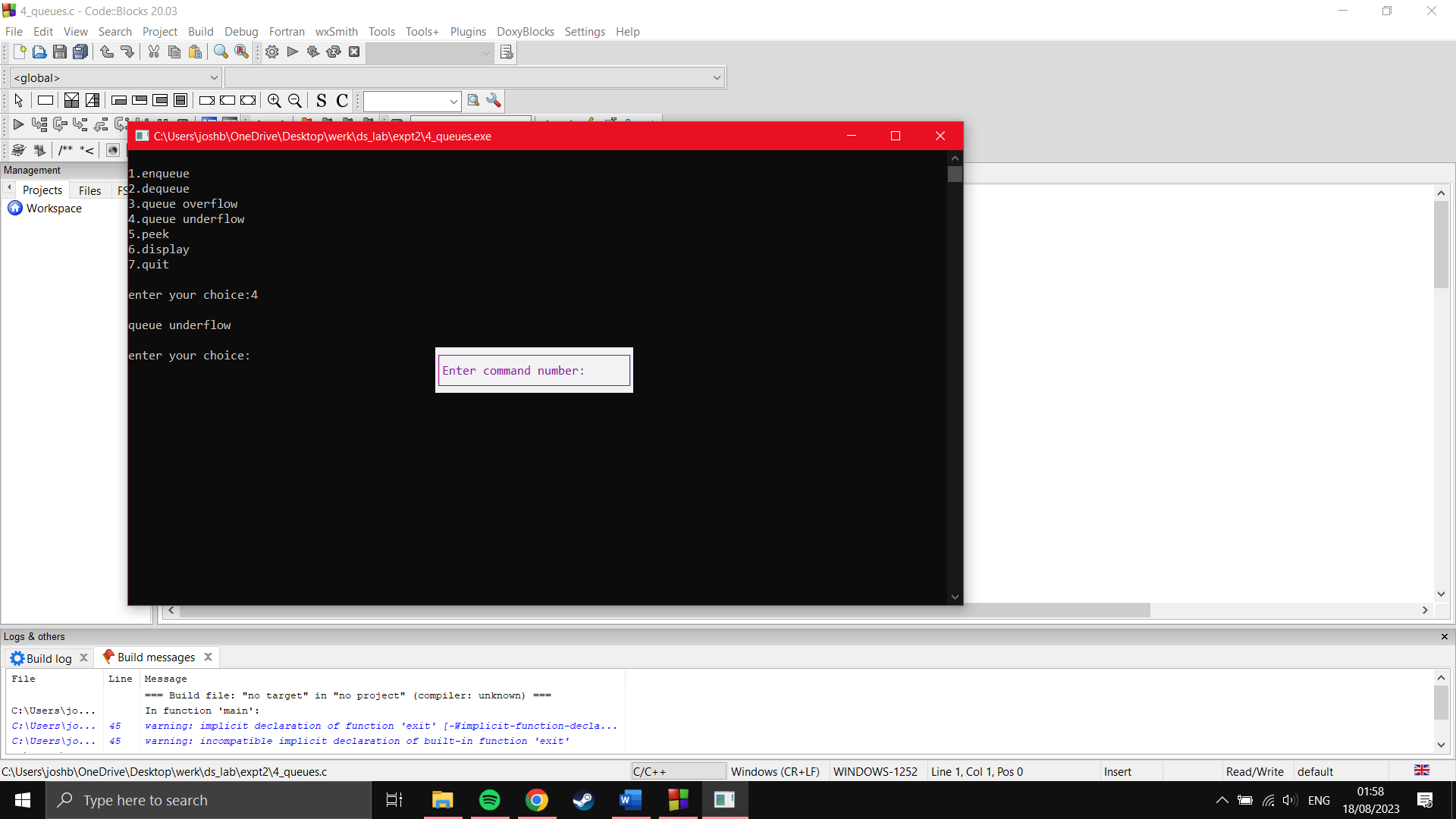


Dequeue:

Expt no: 2



Queue underflow:



Display (for empty queue):

