|  |
| --- |
|  |
| Assignment 2 |
| **Roll no : f17-8195** |
| **M-Usama.**  **Section A** |
| **10/17/2018** |

**Task 4:**

#include<conio.h>

#include<iostream>

using namespace std;

struct stack

{

int d;

stack \*next;

stack \*prev;

};

stack \*head = NULL;

stack \*tail = NULL;

void push(int v) //insert node at end.

{

stack \*n = new stack;

n->d = v;

n->next = NULL;

n->prev = NULL;

if (head == NULL)

{

head = n;

tail = n;

n->prev = head;

}

else

{

tail->next = n;

n->prev = tail;

tail = n;

}

}

void print()

{

stack \*t = head;

while (t!= NULL)

{

cout << t->d<<" ";

t = t->next;

}

cout << endl;

}

void pop() //insertion at end and pop from end.

{

if (head == NULL)

{

cout << "error : there is no element in link list " << endl;

return;

}

if (tail->prev == head)

{

stack \*t=tail;

cout << "poped value : "<<tail->d<<endl;

delete t;

t = NULL;

return;

}

else

{

cout <<"popped value : "<< tail->d<<endl;

stack \*t = tail;

tail = tail->prev;

tail->next = NULL;

delete t;

}

}

int main()

{

push(5);

push(7);

push(8);

push(9);

push(10);

print();

pop();

cout << "after pop one value from link list " << endl;

print();

pop();

cout << "after 2nd pop from link list " << endl;

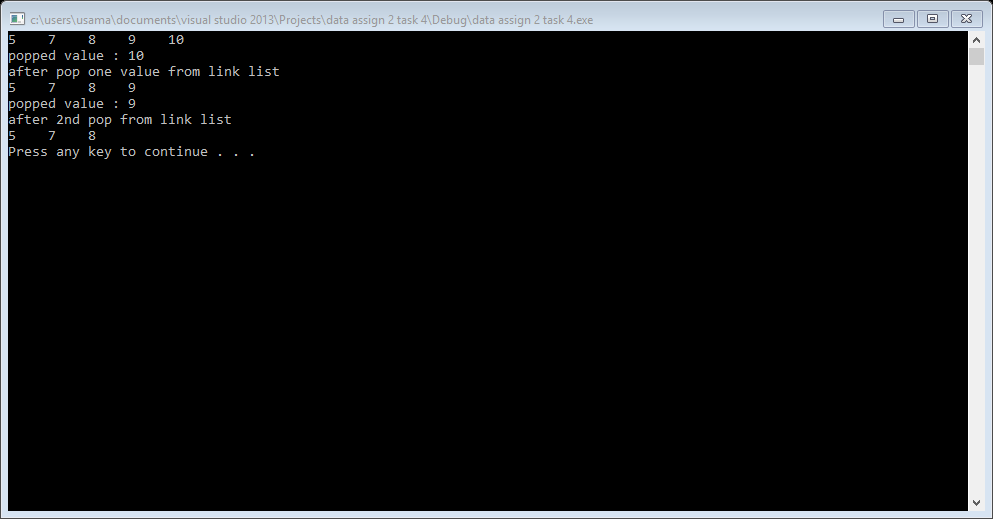
print();

system("pause");

return 0;

}

**Result:**

****

**Task 5:**

#include<iostream>

#include<conio.h>

using namespace std;

class queu

{

int array[10];

int front, rear;

public:

queu()

{

array[10] = { 0 };

front = -1;

rear =-1;

}

bool isempty()

{

return ((rear == -1 && front == -1));

}

bool isfull()

{

return((rear + 1) % 10 == front);

}

void enqueu(int v)

{

if (isfull())

{

cout << "Error : array is full" << endl;

return;

}

if (isempty())

{

front = rear = 0;

}

else

{

rear = (rear + 1) % 10;

}

array[rear] = v;

}

void dequeu()

{

if (isempty())

{

cout << "Error : array is empty " << endl;

return;

}

else if (front == rear)

{

front = -1;

rear = -1;

}

else

{

cout << array[front]<<" : dequeued" << endl;

front = (front + 1) % 10;

}

}

void print()

{

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

{

cout << array[i] << " ";

}

cout << endl;

}

};

int main()

{

queu q;

for (int i = 1; i <= 10; i++)

{

q.enqueu(i);

}

q.print();

q.enqueu(13);

q.dequeu();

for (int i = 0; i < 7; i++)

{

q.dequeu();

}

cout <<"remaing element in queue : ";

q.print();

q.dequeu();

q.dequeu();

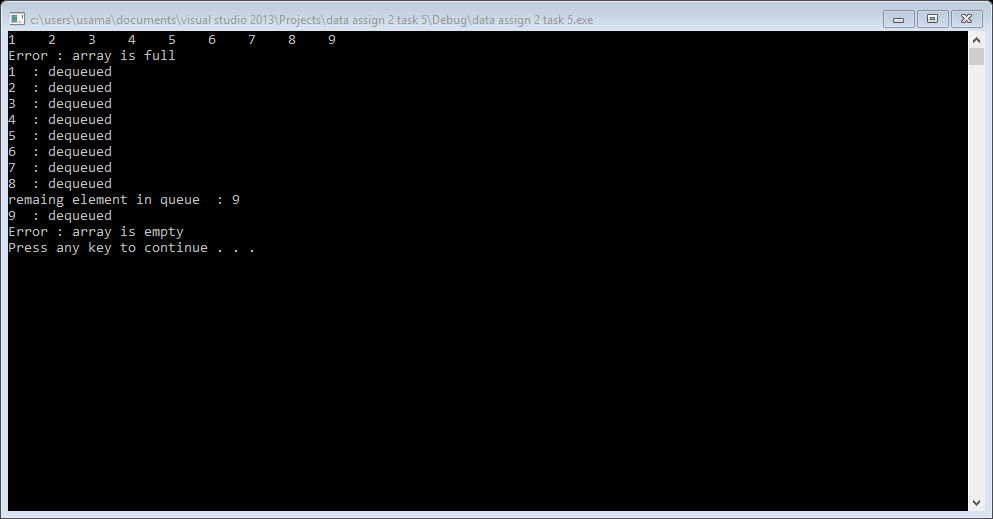
q.dequeu();

system("pause");

return 0;

}

**Result:**



**Task 6:**

#include<iostream>

#include<conio.h>

using namespace std;

class queu

{

int array[10];

int front, rear;

public:

queu()

{

array[10] = { 0 };

front = -1;

rear = -1;

}

bool isempty()

{

return ((rear == -1 && front == -1));

}

bool isfull()

{

return((rear + 1) % 10 == front);

}

void enqueu(int v)

{

if (isfull())

{

cout << "Error : array is full" << endl;

return;

}

if (isempty())

{

front = rear = 0;

}

else

{

rear = (rear + 1) % 10;

}

array[rear] = v;

}

void dequeu()

{

if (isempty())

{

cout << "Error : array is empty " << endl;

return;

}

else if (front == rear)

{

front = -1;

rear = -1;

}

else

{

cout << array[front] << " : dequeued" << endl;

array[front] = { 0 };

front = (front + 1) % 10;

}

}

void print()

{

for (int i = 0; i <10; i++)

{

cout << array[i] << " ";

}

cout << endl;

}

};

int main()

{

queu q;

q.enqueu(25);

q.enqueu(30);

q.enqueu(51);

q.enqueu(60);

q.enqueu(85);

q.enqueu(45);

q.enqueu(88);

q.enqueu(90);

q.enqueu(75);

q.enqueu(95);

q.enqueu(100);

q.print();

q.dequeu();

q.dequeu();

q.dequeu();

cout << "first three spaces is free in over queu, queu now is : " << endl;

q.print();

cout << "now we use first three spaces in over queu " << endl;

q.enqueu(88);

q.enqueu(99);

q.enqueu(100);

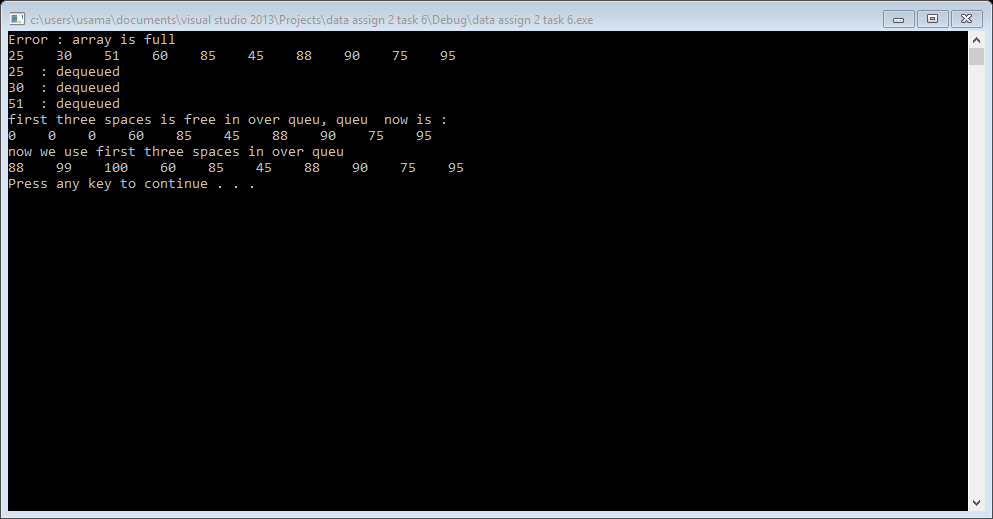
q.print();

system("pause");

return 0;

}

**Result:**

****

**task 7:**

#include<iostream>

#include<conio.h>

using namespace std;

struct queu

{

int data;

queu \*next;

};

queu \*head = NULL;

queu \*tail = NULL;

void enqueu(int v) //insertion at end and dequeu at head

{

queu \*n = new queu;

n->next = NULL;

n->data = v;

if (head == NULL)

{

head = n;

tail = n;

}

else

{

tail->next = n;

tail = n;

}

}

void dequeu()

{

if (head == NULL)

{

cout << "Error : no element exit in queu" << endl;

return;

}

else

{

queu \*t = head;

cout << t->data<<" : dequeud value "<<endl;

head = head->next;

delete t;

t = NULL;

}

}

void print()

{

queu \*t = head;

while (t != NULL)

{

cout << t->data << " ";

t = t->next;

}

cout << endl;

}

int main()

{

queu q;

enqueu(1);

enqueu(2);

enqueu(3);

cout << "we insert following elements : " << endl;

print();

dequeu();

cout << "now over queu is :" << endl;

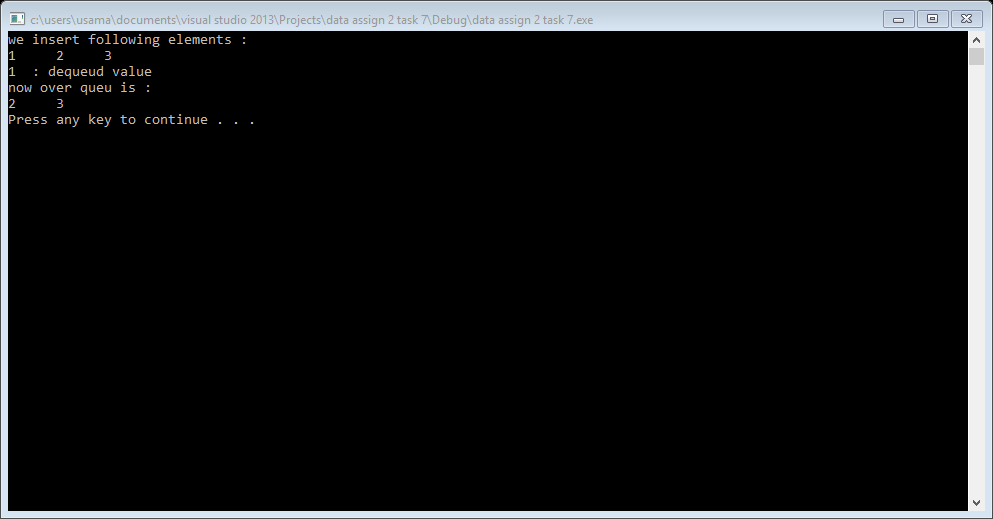
print();

system("pause");

return 0;

}

**result:**

****