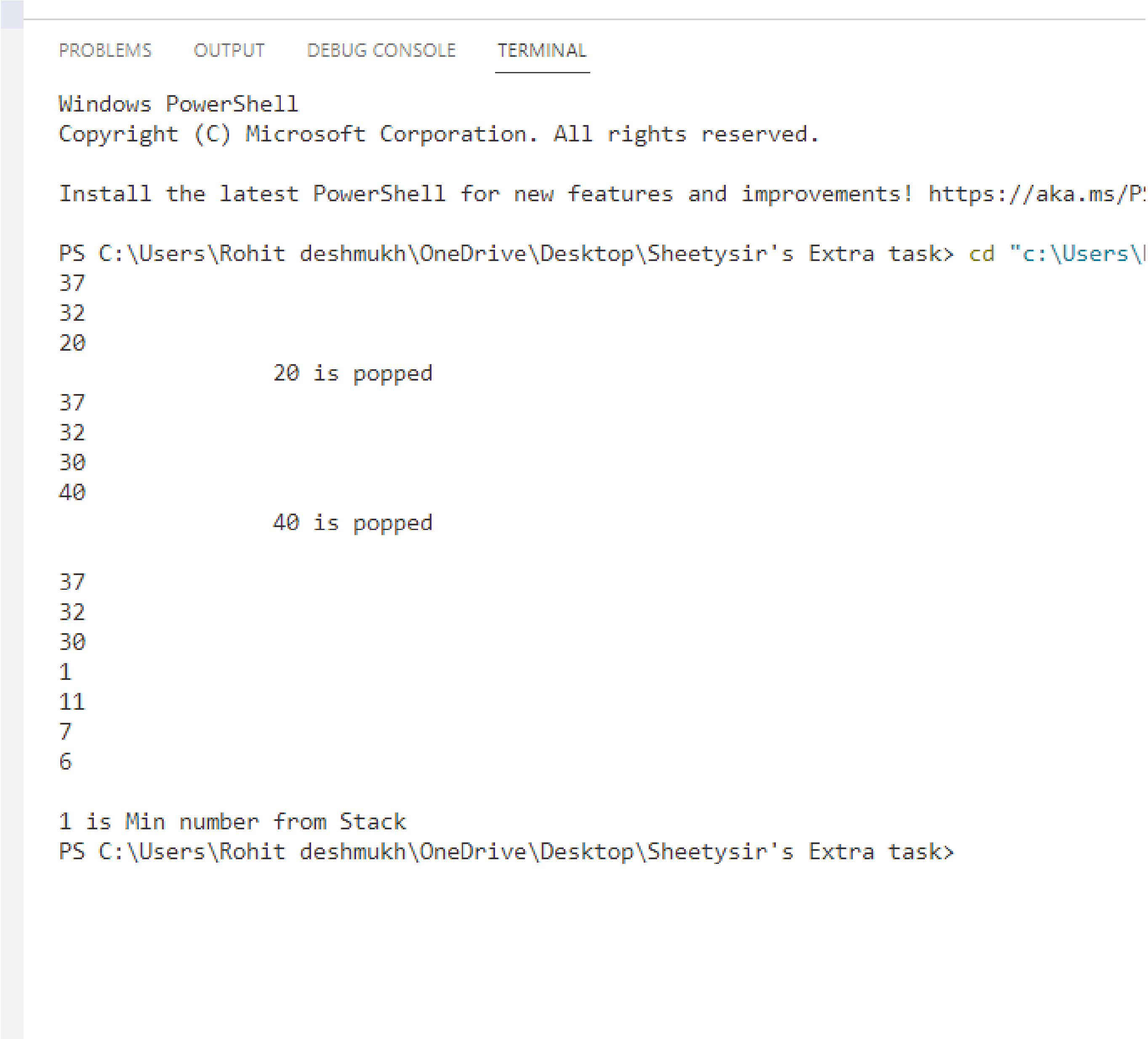
Avantika Shailesh Deshmukh
2021BIT153

Q.Get minimum element from stack :

```
#include <stdio.h>
#define size 10
int top = -1;
int arr[size];

void push(int x);
void pop();
void show();
void getmin()
{
 int Q = arr[0];
 for (int i = 0; i <= top; i++)
 {

  if (Q > arr[i])
  {
   Q = arr[i];
  }
 }
 printf("%d is Min number from Stack\n", Q);
}
int main()
{
 push(37);
 push(32);
 push(20);
 show();
 printf("\t\t");
 pop();

 push(30);
 push(40);

 show();
 printf("\t\t");
 pop();
 push(1);
 push(11);
 push(7);
 push(6);

 printf("\n");

 show();

 printf("\n");

 getmin();
}
void push(int x)
{

 if (x != -1)
 {

  ++top;
  arr[top] = x;
 }
 else if (top == -1)
 {
  printf("Stack is empty\n");
  return;
 }
 else if (top == size - 1)
 {
  printf("Overflow\n");
 }
 else
  return;
}
void pop()
{
 if (top != -1)
 {
  printf("%d is popped\n", arr[top]);
  top = top - 1;
 }
 else if (top == -1)
 {
  printf("Underflow\n");
 }
}
void show()
{
 for (int i = 0; i <= top; i++)
 {
  printf("%d\n", arr[i]);
 }
}
```

Output:

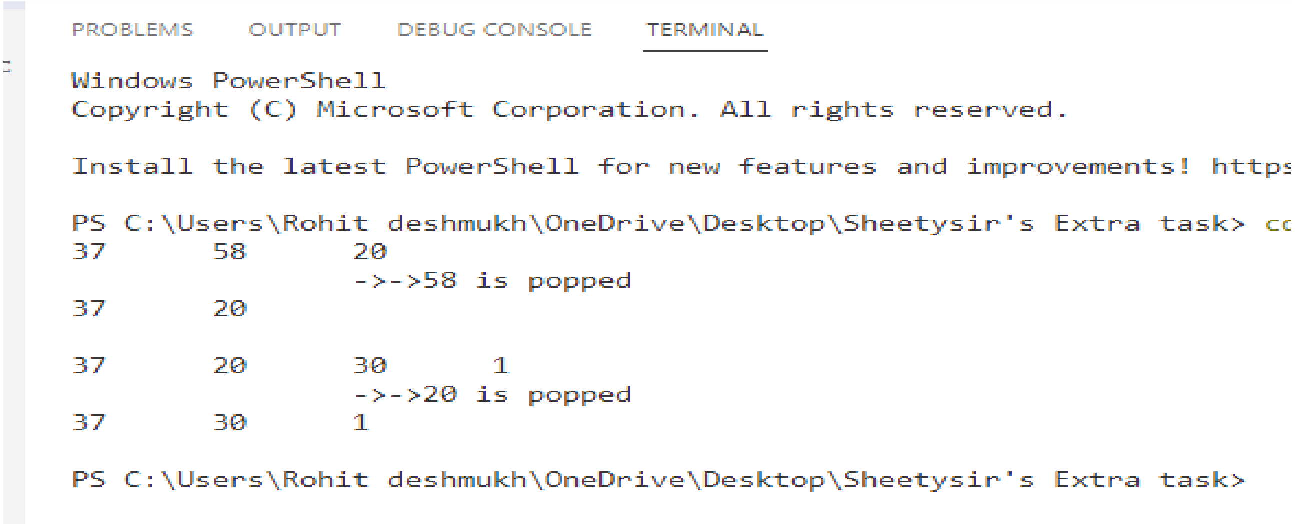


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/P

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task> cd "c:\Users\
37
32
20
                20 is popped
37
32
30
40
                40 is popped

37
32
30
1
11
7
6

1 is Min number from Stack
PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task>
```

Q.Delete middle element of stack:

```
#include <stdio.h>
#define size 10
int top = -1;
int arr[size];

void push(int x);
void pop();
void show();

int main()
{
 push(37);
 push(58);
 push(20);
 show();
 printf("\t\t");
 pop();
 show();
 printf("\n\n");
 push(30);
 push(1);

 show();
 pop();
 show();
 printf("\n\n");
}
void push(int x)
{

 if (x != -1)
 {

  ++top;
  arr[top] = x;
 }
 else if (top == -1)
 {
  printf("Stack is empty\n");
  return;
 }
 else if (top == size - 1)
 {
  printf("Overflow\n");
 }
 else
  return;
}
void pop()
{
 if (top != -1)
 {
  printf("\n\t\t->->%d is popped\n", arr[top / 2]);
  arr[top / 2] = arr[(top / 2) + 1];

  for (int i = ((top / 2) + 1); i < top; i++)
  {
   arr[i] = arr[top];
  }
  top = top - 1;
 }
 else if (top == -1)
 {
  printf("Underflow\n");
 }
}
void show()
{
 for (int i = 0; i <= top; i++)
 {
  printf("%d\t", arr[i]);
 }
}
```

Output:



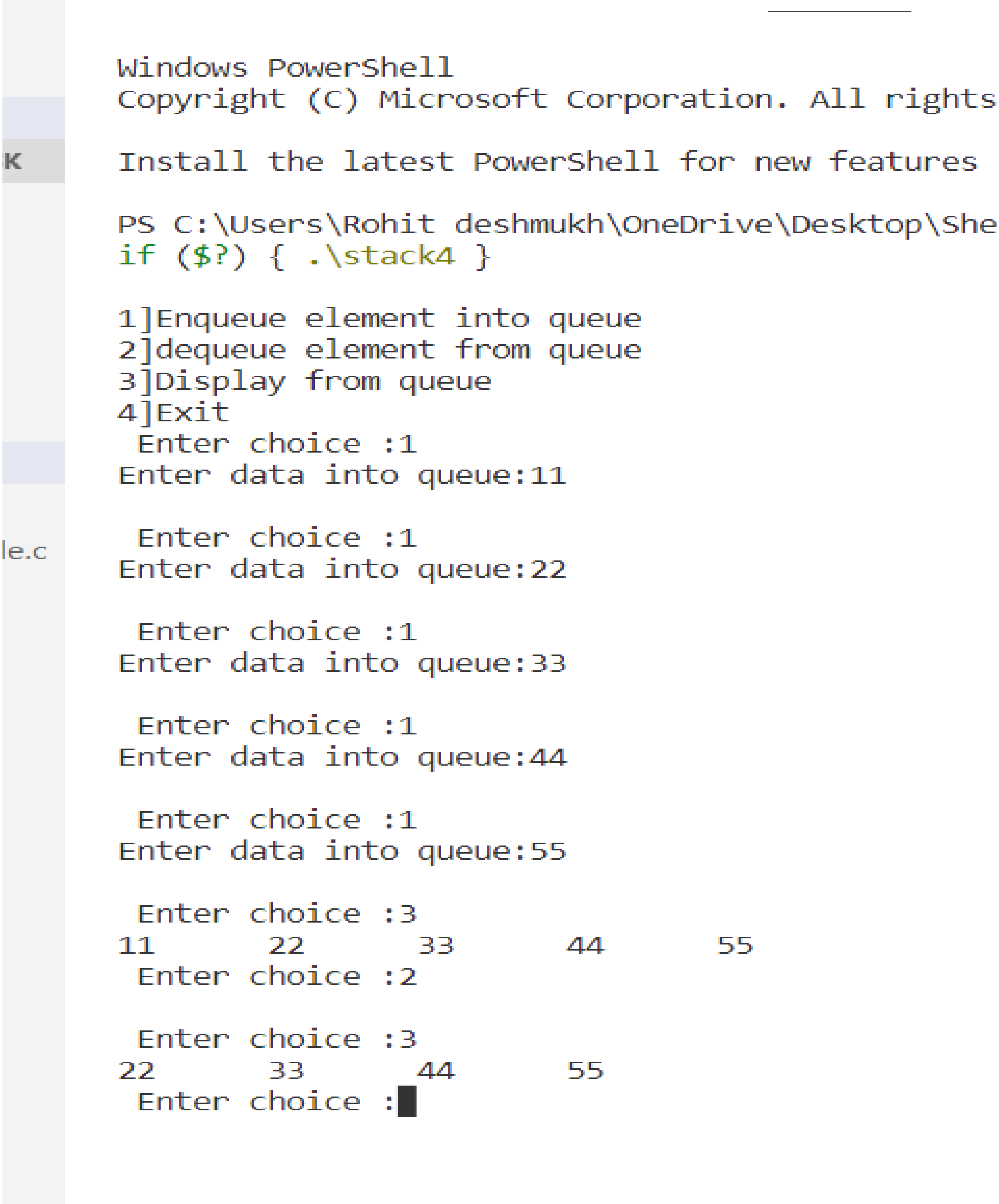
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task> cc
37      58      20
                ->->58 is popped
37      20

37      20      30      1
                ->->20 is popped
37      30      1

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task>
```

Q. Enqueue Dequeue using stack:

```
#include <stdio.h>
#include <stdlib.h>
void push1(int);
void push2(int);
int pop1();
int pop2();
void enqueue();
void dequeue();
void display();
void create();

int s1[50], s2[50];
int top1 = -1, top2 = -1;
int count = 0;
void main()
{
 int ch;
 printf("\n1]Enqueue element into queue");
 printf("\n2]dequeue element from queue");
 printf("\n3]Display from queue");
 printf("\n4]Exit");
 create();
 while (1)
 {
  printf("\n Enter choice :");
  scanf("%d", &ch);
  switch (ch)
  {
  case 1:
   enqueue();
   break;
  case 2:
   dequeue();
   break;
  case 3:
   display();
   break;
  case 4:
   exit(0);
  default:
   printf("Enter valid choice\n");
  }
 }
}
void create()
```

```
{
 top1 = top2 = -1;
}
void push1(int data)
{
 s1[++top1] = data;
}
int pop1()
{
 return (s1[top1--]);
}
void push2(int data)
{
 s2[++top2] = data;
}
int pop2()
{
 return (s2[top2--]);
}

void enqueue()
{
 int data, i;
 printf("Enter data into queue:");
 scanf("%d", &data);
 push1(data);
 count++;
}
void dequeue()
{

 for (int i = 0; i <= count; i++)
 {
  push2(pop1());
 }
 pop2();
 count--;
 for (int i = 0; i <= count; i++)
 {
  push1(pop2());
 }
}
void display()
{
 for (int i = 0; i <= top1; i++)
 {
  printf("%d\t", s1[i]);
 }
}
```

Output:

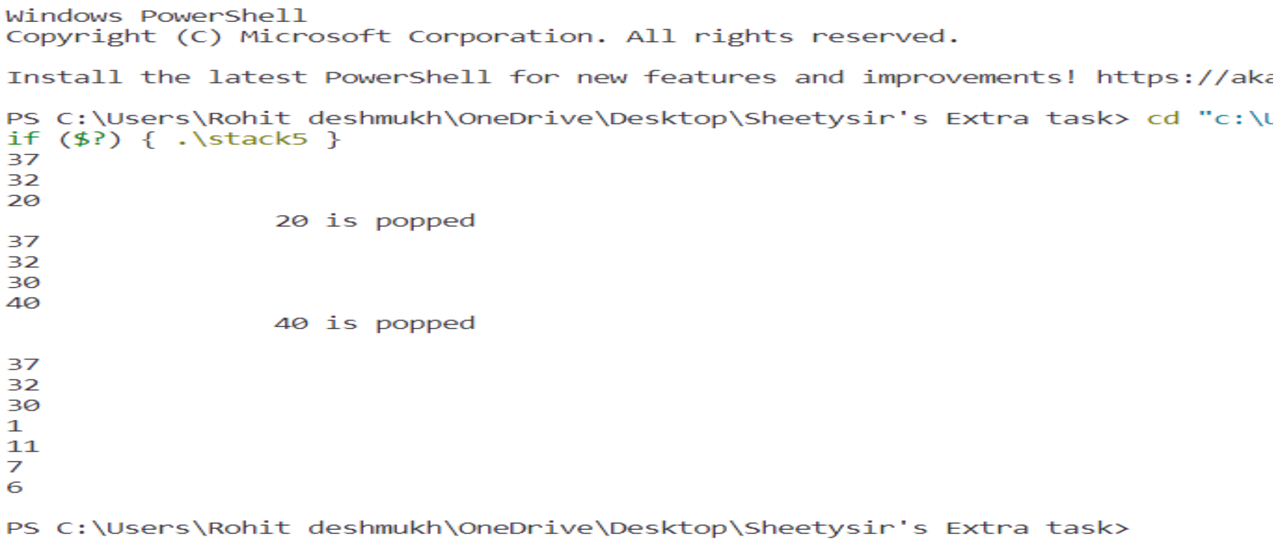


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights

Install the latest PowerShell for new features

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\She
if ($?) { .\stack4 }

1]Enqueue element into queue
2]dequeue element from queue
3]Display from queue
4]Exit
 Enter choice :1
Enter data into queue:11

 Enter choice :1
Enter data into queue:22

 Enter choice :1
Enter data into queue:33

 Enter choice :1
Enter data into queue:44

 Enter choice :1
Enter data into queue:55

 Enter choice :3
11      22      33      44      55
 Enter choice :2

 Enter choice :3
22      33      44      55
 Enter choice :
```

Q.Stack operation using array:

```
#include <stdio.h>

#define size 10
int top = -1;
int arr[size];

void push(int x);
void pop();
void show();
int main()
{
 push(37);
 push(32);
 push(20);
 show();
 printf("\t\t");
 pop();

 push(30);
 push(40);

 show();
 printf("\t\t");
 pop();
 push(1);
 push(11);
 push(7);

 push(6);

 printf("\n");

 show();

 printf("\n");
}
void push(int x)
{

 if (x != -1)
 {

  ++top;
  arr[top] = x;
 }
 else if (top == -1)
 {
  printf("Stack is empty\n");
  return;
 }
 else if (top == size - 1)
 {
  printf("Overflow\n");
 }
 else
  return;
}
void pop()
{
 if (top != -1)
 {
  printf("%d is popped\n", arr[top]);
  top = top - 1;
 }
 else if (top == -1)
 {
  printf("Underflow\n");
 }
}
void show()
{
 for (int i = 0; i <= top; i++)
 {
  printf("%d\n", arr[i]);
 }
}
```

Output:



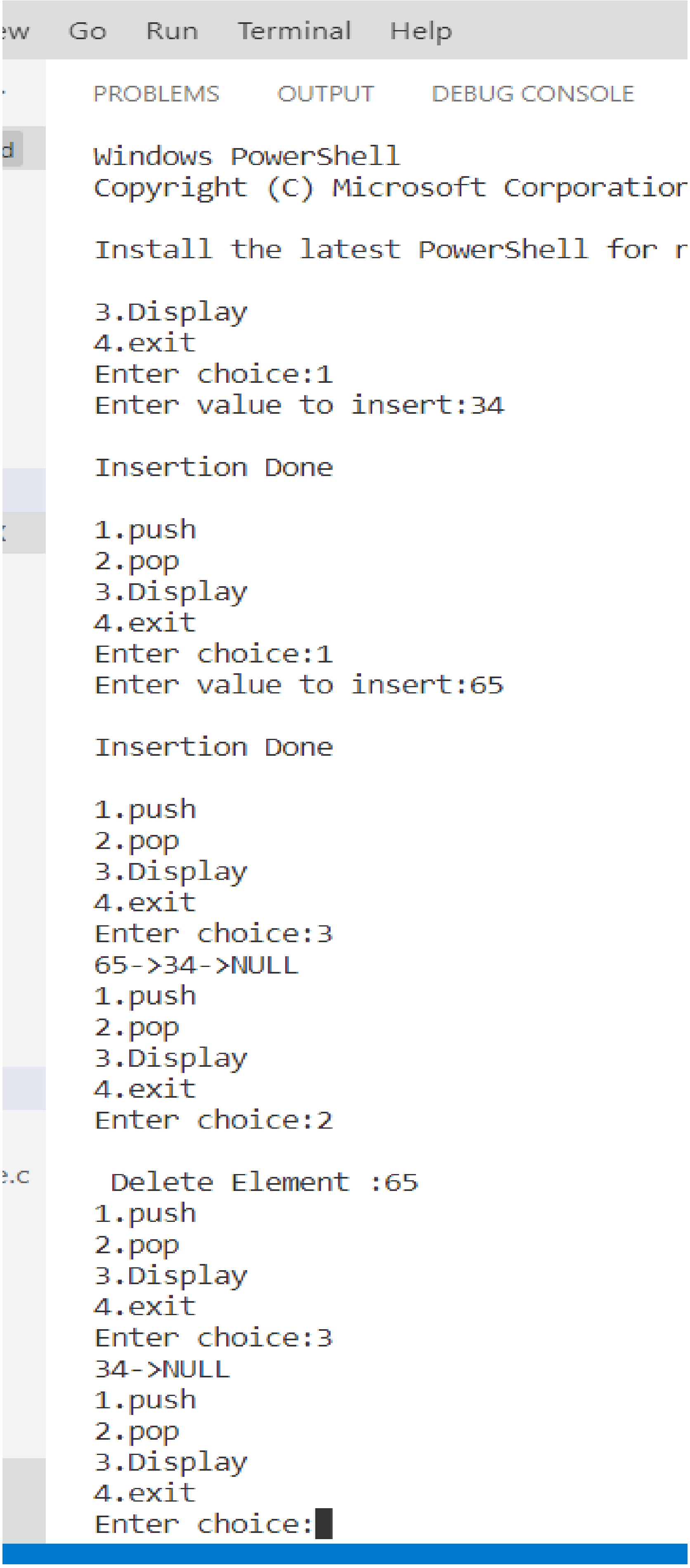
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task> cd "c:\U
if ($?) { .\stack5 }
37
32
20
                20 is popped
37
32
30
40
                40 is popped

37
32
30
1
11
7
6

PS C:\Users\Rohit deshmukh\OneDrive\Desktop\Sheetysir's Extra task>
```

Q.Stack using Linked List

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

struct node
{
 int data;
 struct node *next;

} *top = NULL;

void push(int);
void pop();
void Display();
void main()
{
 int choice, value;
 printf("Stack Using Linked List:");
 while (1)
 {
  printf("\n1.push \n2.pop \n3.Display \n4.exit");
  printf("\nEnter choice:");
  scanf("%d", &choice);
  switch (choice)
  {
  case 1:
   printf("Enter value to insert:");
   scanf("%d", &value);
   push(value);
   break;
  case 2:
   pop();
   break;
  case 3:
   Display();
   break;
  case 4:
   exit(0);
   break;

  default:
   printf("Enter valid ");
  }
 }
}
```

```
void push(int value)
{
 struct node *newnode;
 newnode = (struct node *)malloc(sizeof(struct node*));
 newnode->data = value;
 if (top == NULL)
 {
  newnode->next = NULL;
 }
 else
  newnode->next = top;
 top = newnode;
 printf("\nInsertion Done\n");
}
void pop()
{
 if (top == NULL)
  printf("\nStack Is empty ");
 else
 {
  struct node *temp = top;
  printf("\n Delete Element :%d", temp->data);
  top = temp->next;
  free(temp);
 }
}
void Display()
{
 if (top == NULL)
  printf("\n Stack is Empty");
 else
 {
  struct node *temp = top;
  while (temp->next != NULL)
  {
   printf("%d->", temp->data);
   temp = temp->next;
  }
  printf("%d->NULL", temp->data);
 }
}
```

Output:



```
Windows PowerShell
Copyright (C) Microsoft Corporatior

Install the latest PowerShell for r

3.Display
4.exit
Enter choice:1
Enter value to insert:34

Insertion Done

1.push
2.pop
3.Display
4.exit
Enter choice:1
Enter value to insert:65

Insertion Done

1.push
2.pop
3.Display
4.exit
Enter choice:3
65->34->NULL
1.push
2.pop
3.Display
4.exit
Enter choice:2

 Delete Element :65
1.push
2.pop
3.Display
4.exit
Enter choice:3
34->NULL
1.push
2.pop
3.Display
4.exit
Enter choice:
```