ARYAMAN MISHRA 19BCE1027

1) #include<stdio.h>

#include<cstdlib>

struct Node {

int data;

struct Node\* link;

};

struct Node\* top;

void push(int data)

{

struct Node\* temp;

temp = new Node();

if (!temp) {

printf("\nHeap Overflow");

exit(1);

}

temp->data = data;

temp->link = top;

top = temp;

}

int isEmpty()

{

return top == NULL;

}

int peek()

{

if (!isEmpty())

return top->data;

else

exit(1);

}

void pop()

{

struct Node\* temp;

if (top == NULL)

{

printf("\nStack Underflow");

exit(1);

}

else

{

temp = top;

top = top->link;

temp->link = NULL;

free(temp);

}

}

void display()

{

struct Node\* temp;

if (top == NULL) {

printf("\nStack Underflow");

exit(1);

}

else {

temp = top;

while (temp != NULL)

{

printf("\n%d",temp->data);;

temp = temp->link;

}

}

}

int main()

{

int choice=1,n,ch;

while(choice==1)

{

printf("\nStack Menu");

printf("\n\n1.Push\n2.Pop\n3.Display\n4.View Top Element\n5.Exit.");

printf("\n\nEnter your choice(1-5):");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("Enter an element in stack.\n");

scanf("%d",&n);

push(n);

break;

case 2: pop();

break;

case 3: display();

break;

case 4: printf("\nTop element is %d\n",peek());

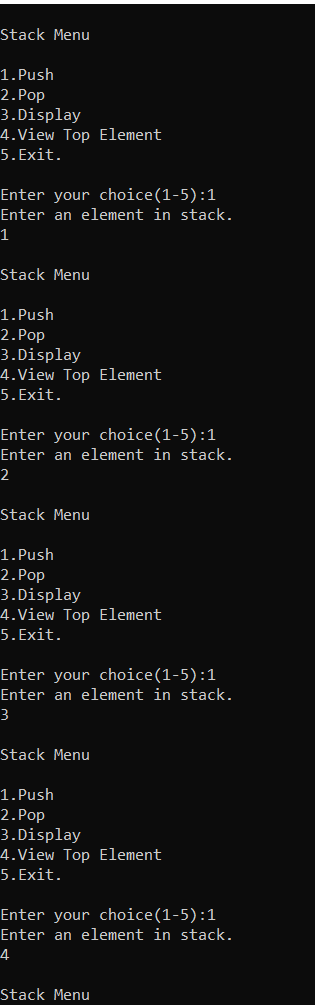
case 5:exit(1);

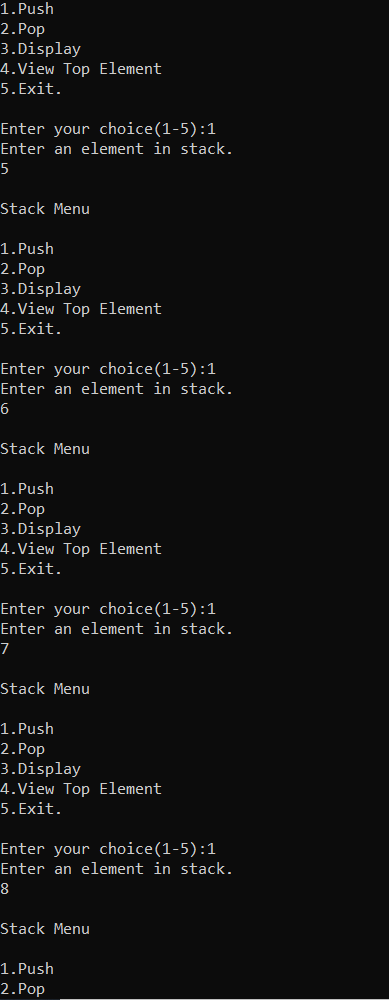
default: printf("\nWrong Choice!!");

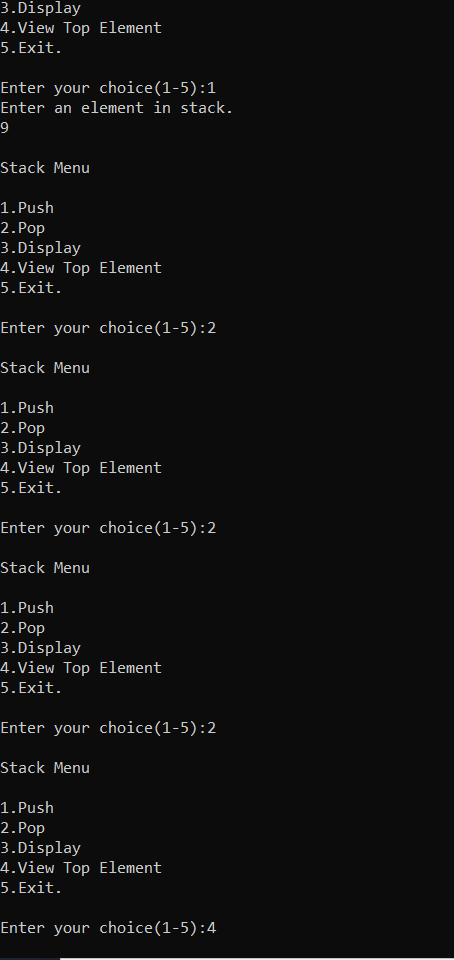
}

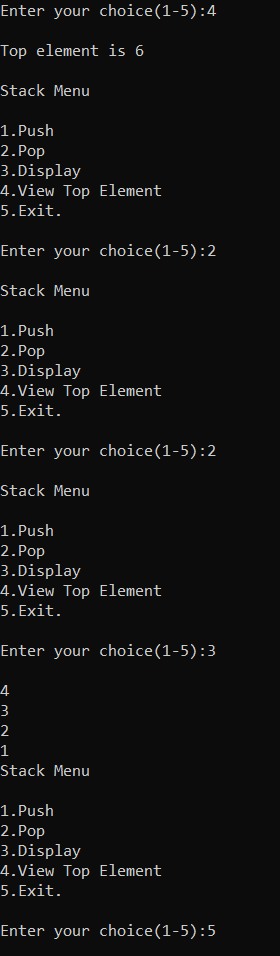
}

}









2)#include <cstdlib>

#include <stdio.h>

struct Node {

int data;

struct Node\* next;

};

int count=0;

int largestElement(struct Node\* head)

{

int max = INT\_MIN;

while (head != NULL) {

if (max < head->data)

max = head->data;

head = head->next;

}

return max;

}

int smallestElement(struct Node\* head)

{

int min = INT\_MAX;

while (head != NULL) {

if (min > head->data)

min = head->data;

head = head->next;

}

return min;

}

void push(struct Node\*\* head, int data)

{

struct Node\* newNode =

(struct Node\*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = (\*head);

(\*head) = newNode;

count++;

}

void printList(struct Node\* head)

{

while (head != NULL)

{

printf("%d -> ", head->data);

head = head->next;

}

printf("NULL\n" );

}

void deleteNode(struct Node \*\*head\_ref, int position)

{

if (\*head\_ref == NULL)

return;

struct Node\* temp = \*head\_ref;

if (position == 0)

{

\*head\_ref = temp->next; // Change head

free(temp); // free old head

return;

}

for (int i=0; temp!=NULL && i<position-1; i++)

temp = temp->next;

if (temp == NULL || temp->next == NULL)

return;

struct Node \*next = temp->next->next;

free(temp->next);

temp->next = next;

count--;

}

int main()

{

struct Node\* head = NULL;

int choice=1,n,ch;

while(choice==1)

{

printf("\nStack Menu");

printf("\n\n1.Push\n2.Pop\n3.Display\n4.Print maximum element.\n5.Print minimum element.\n6.Print Number of Nodes.\n7.Exit Program.");

printf("\n\nEnter your choice(1-7):");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("Enter an element in stack.\n");

scanf("%d",&n);

push(&head,n);

break;

case 2: printf("Enter position at which element has to be deleted.\n");

scanf("%d",&n);

deleteNode(&head,n);

break;

case 3: printList(head);

break;

case 4: printf("Maximum element in linked list:\n");

printf("%d\n",largestElement(head));

break;

case 5: printf("Minimum element in linked list:\n");

printf("%d\n",smallestElement(head));

break;

case 6: printf("Number of nodes:\n");

printf("%d",count);

break;

case 7: exit(1);

default: printf("\nWrong Choice!!");

}

}

}