```
#include <iostream>
using namespace std;

#define MAX 10

struct stack {
  int items[MAX];
  int top;
};
typedef struct stack st;

void createEmptyStack(st *s) {
  s->top = -1;
}

int isFull(st *s) {
  if (s->top == MAX - 1)
    return 1;
  else
```

```
    return 0;
}

int isEmpty(st *s) {
  if (s->top == -1)
    return 1;
  else
    return 0;
}

void push(st *s, int newitem) {
  if (isFull(s)) {
    cout << "STACK FULL";
  } else {
    s->top++;
    s->items[s->top] = newitem;
  }
}
```

```
void pop(st *s) {
  if (isEmpty(s)) {
    cout << "\n STACK EMPTY \n";
  } else {
    cout << "Item popped= " << s->items[s->top];
    s->top--;
  }
  cout << endl;
}

void printStack(st *s) {
  printf("Stack: ");
  for (int i = 0; i <= s->top; i++) {
    cout << s->items[i] << " ";
  }
  cout << endl;
}

int main() {
```

```
int ch, item;
char repeat;
st *s = (st *)malloc(sizeof(st));
createEmptyStack(s);

do {
  cout << "1. Pop" << endl;
  cout << "2. Push" << endl;
  cout << "3. Display" << endl;
  cout << "4. Exit" << endl;
  cout << "Enter your choice: ";
  cin >> ch;

  switch (ch) {
    case 1:
      pop(s);
      break;
    case 2:
      cout << "Enter value to be pushed:";
```

```
      cin >> item;
      push(s, item);
      break;
    case 3:
      printStack(s);
      break;
    case 4:
      exit(0);
    default:
      cout << "Invalid Choice" << endl;
   }

   cout << "Do you want to continue? (y/n): ";
   cin >> repeat;
  } while (repeat == 'y' || repeat == 'Y');

  return 0;
}
```