DATA STRUCTURES AND ALGORITHM

Ex 4: STACKS

Name: A Hari Krishna Swarun

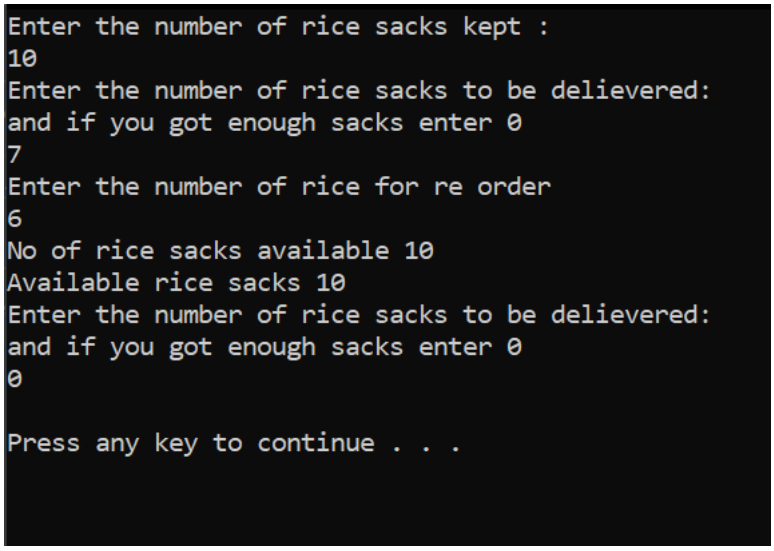
Reg no:21BPS1252

Q. Consider a retailer shop, which has the rack with the capacity to place 10 sacks of rice. The store keeper can place the sack only at the top as well as take the sack only from the top. The store keeper is not permitted to take the sack from the middle. Further the reorder level is 4. If the rack reaches the reorder level then the system should raise an exception to reorder the rice of sacks. Implement the above scenario using an appropriate data.

```c
#include<stdio.h>
#include<stdlib.h>
#define size 10

struct stack{
  int stock[size];
  int top;
}rice;

void full(int item){
  if(rice.top==(size-1)){
    printf("rice sacks are full in stack\n");
  }
  else{
    rice.top=rice.top+1;
    rice.stock[rice.top]=item;
  }
}

void empty(){
  if(rice.top==-1){
    printf("Rice sacks are full in the stack\n");
  }
  else{
    rice.top=rice.top-1;
  }
}
```

```
int main(){
  int n;
  rice.top=-1;
  printf("Enter the number of rice sacks kept :\n");
  scanf("%d",&n);
  for(int i=0;i<n;i++){
    full(1);
  }
  int input=1;

  while(input!= 0){
    printf("Enter the number of rice sacks to be delievered:\nand if you got enough sacks enter 0\n");
    scanf("%d",&input);
    if(input==0){
      break;
    }
    else{
      for(int i=0;i<input;i++){
        if(rice.top==3){
          printf("Enter the number of rice for re order\n");
          scanf("%d",&n);
          for(int i=0;i<n;i++){
            full(1);
          }
          printf("No of rice sacks available %d\n",rice.top+1);
        }
        else{
          }
          else{
            empty();
          }
        }
        printf("Available rice sacks %d\n",rice.top+1);
      }
    }
    return 0;
}
```

OUTPUT:


Enter the number of rice sacks kept :
10
Enter the number of rice sacks to be delievered:
and if you got enough sacks enter 0
7
Enter the number of rice for re order
6
No of rice sacks available 10
Available rice sacks 10
Enter the number of rice sacks to be delievered:
and if you got enough sacks enter 0
0
Press any key to continue . . .

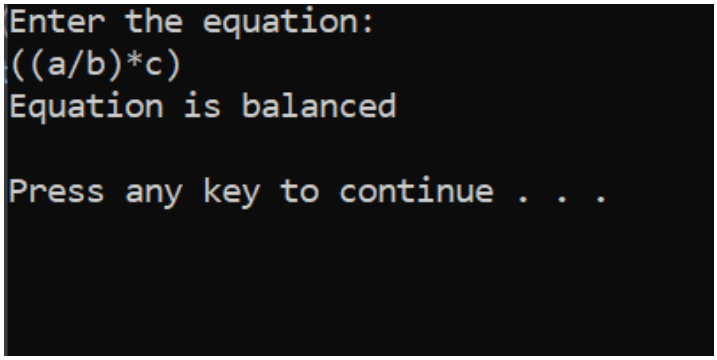
Q2. Using the concept of stack check whether the parenthesis in a given expression is balanced or not. For example, in the expression ((a+b)-(c)) the number of right parenthesis (3) is equal to the left parenthesis (3).So the expression is said to balanced. If not, the expression is said to be unbalanced.

CODE:

```c
#include<stdio.h>
#define size 100
int top=-1;
char ex[100];
int valid(char ch[]){
  for(int i=0;ch[i]!='\0';i++){
    if(top==size-1){
      printf("Stack overflow");
      return 0;
    }
    if(ch[i]=='('
    ex[++top]=ch[i];
    if(ch[i]==')'){
      if(top==-1)
      return 0;
      if(ex[top--]!='('
      return 0;
    }
  }
  return -1;
}
```

```c
int main(){
  char ar[100];
  printf("Enter the equation:\n");
  scanf("%[^\n]s",ar);
  if(valid(
  ar))
  printf("Equation is balanced\n");
  else
  printf("Equation is not balanced\n");
  return 0;
}
```

OUTPUT:


Enter the equation:
((a/b)*c)
Equation is balanced
Press any key to continue . . .