Task 6:

The idea of code:

1. The user input process.
2. Entry is possible to be an open arc or arc closed or the process of collecting, subtract, multiply or divide .. Is claimed that he considers a number.
3. Will be stored in input variables when the input arc is being closed in order to process ... Required between the brackets (The same idea of the algorithm).

Program:

```
#include<iostream.h>
#include<malloc.h>
#include<string.h>
struct stack {
char n;
struct stack *next;
};
typedef struct stack *STK;
typedef STK STACK;

STACK push(STACK head, char n)
{
STACK temp=(stack*)malloc(sizeof(struct stack));
if (temp==NULL){
cout<<"error,no space \n";
}
else
{
temp->n=n;
temp->next=head;
head=temp;
}
return head;
}
```

```
char pop(STACK head)
{
if (head==NULL)
{
cout<<"error \n";
}
else
{
STACK top=head;
char d;
d=top->n;
head=top->next;
free(top);
return d;
}
return 0;
}
void main()
{
char x;
STACK head;
head=NULL;
s: cout<<"enter charecter in list or n to exit\n";
 cin>>x;
if(x == 'n'){
 goto e;
}
if(x == '('){
head=push(head,'(');
goto s;
}
else if (x == '+')
{
head=push(head,'+');
goto s;
}
else if (x == '-'){
head=push(head,'-');
goto s;
}
else if (x == '*'){
head=push(head,'*');
goto s;
```

```
}
else if (x == '/'){
head=push(head,'/');
goto s;
}
else if (x == ')'){
char op1,op2,op;
int t=0;
op1=pop(head);
cout<<op1<<endl;
op=pop(head);
cout<<op<<endl;
op2=pop(head);
cout<<op2<<endl;
int p1= int(op1);
cout<<"p1"<<p1<<endl;
int p2=int(op2);
cout<<"p2"<<p2<<endl;
if (op == '+'){
t=p1+p2;
}
if (op == '-'){
t=p1-p2;
}
if (op == '*'){
t=p1*p2;
}
if (op == '/'){
t=p1 / p2;
}
pop(head);
cout<<"anser:  "<<t<<endl;
char r=char (t);
head=push(head,r);
goto s;
 }
else {
head=push(head,x);
goto s;
}
e: cout<<"good bay";
 }
```