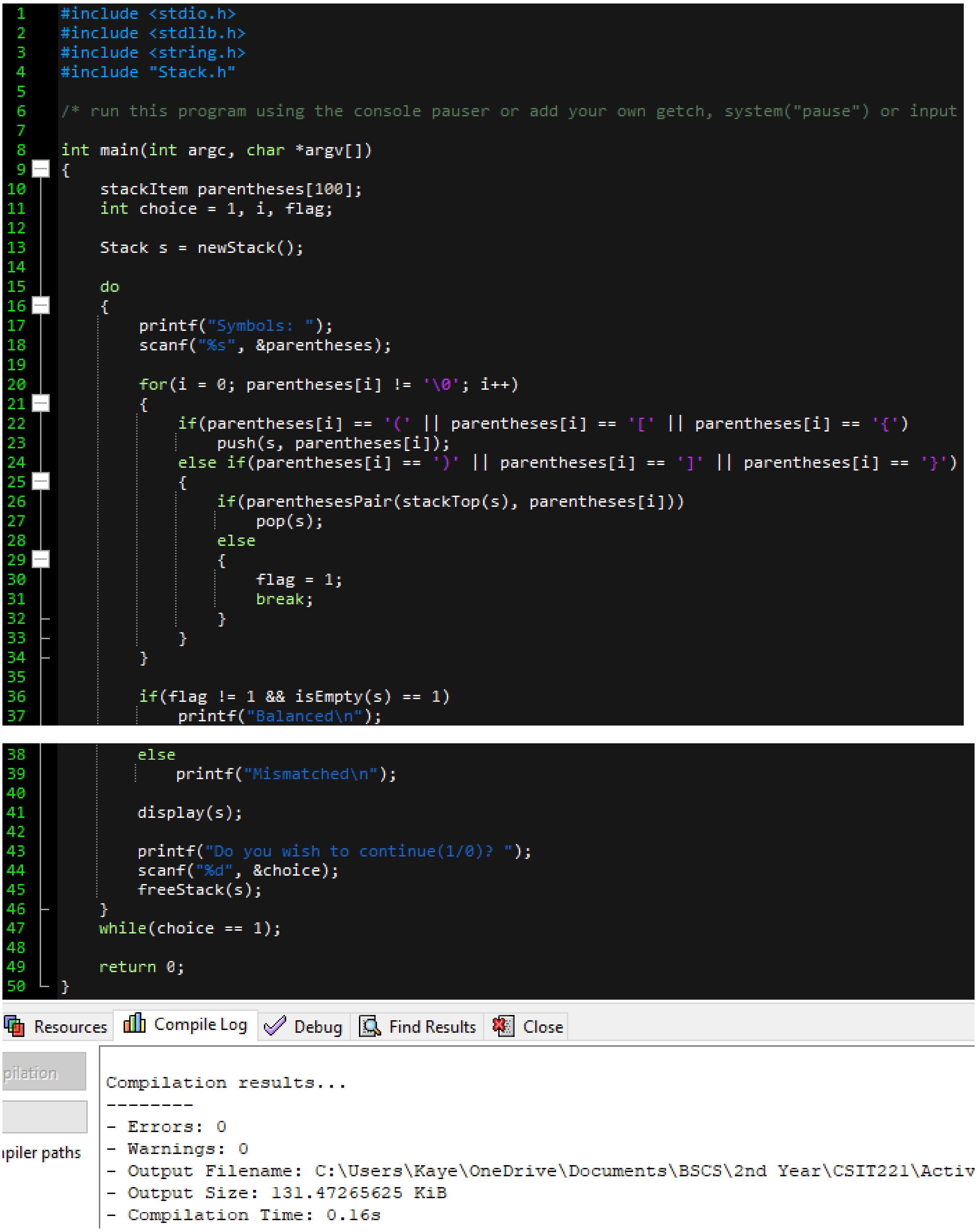
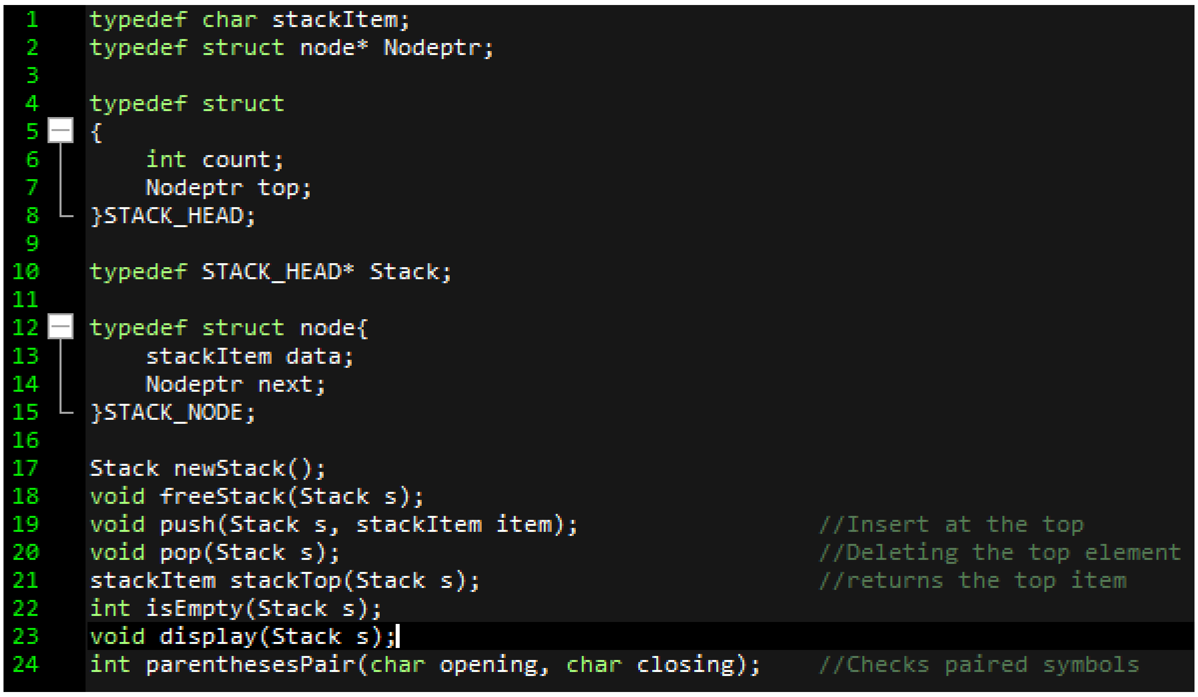


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Stack.h"

/* run this program using the console pauser or add your own getch, system("pause") or input

int main(int argc, char *argv[])
{
    stackItem parentheses[100];
    int choice = 1, i, flag;

    Stack s = newStack();

    do
    {
        printf("Symbols: ");
        scanf("%s", &parentheses);

        for(i = 0; parentheses[i] != '\0'; i++)
        {
            if(parentheses[i] == '(' || parentheses[i] == '[' || parentheses[i] == '{')
                push(s, parentheses[i]);
            else if(parentheses[i] == ')' || parentheses[i] == ']' || parentheses[i] == '}')
            {
                if(parenthesesPair(stackTop(s), parentheses[i]))
                    pop(s);
                else
                {
                    flag = 1;
                    break;
                }
            }
        }

        if(flag != 1 && isEmpty(s) == 1)
            printf("Balanced\n");
        else
            printf("Mismatched\n");

        display(s);

        printf("Do you wish to continue(1/0)? ");
        scanf("%d", &choice);
        freeStack(s);
    }
    while(choice == 1);

    return 0;
}
```

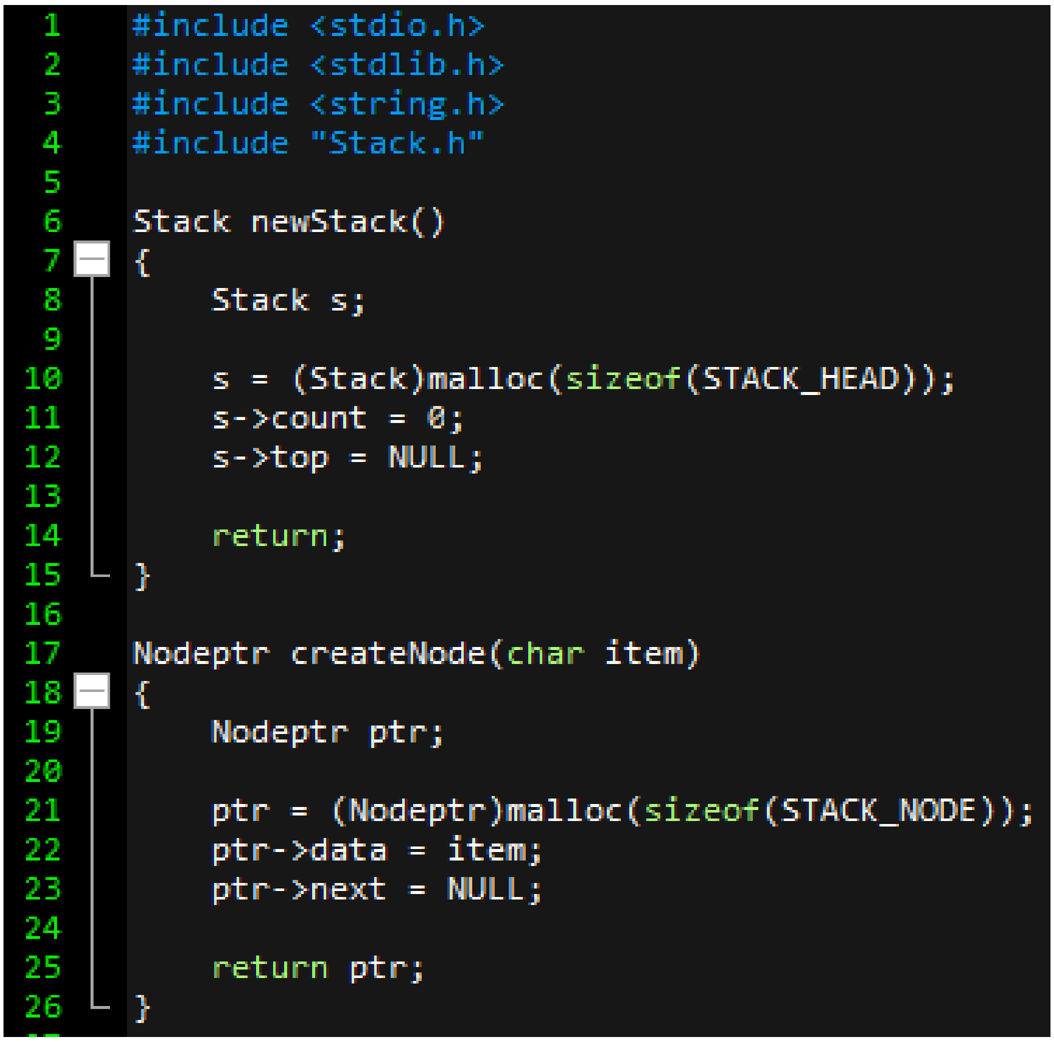
```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Kaye\OneDrive\Documents\BSCS\2nd Year\CSIT221\Activ
- Output Size: 131.47265625 KiB
- Compilation Time: 0.16s
```

Figure 1. main.c

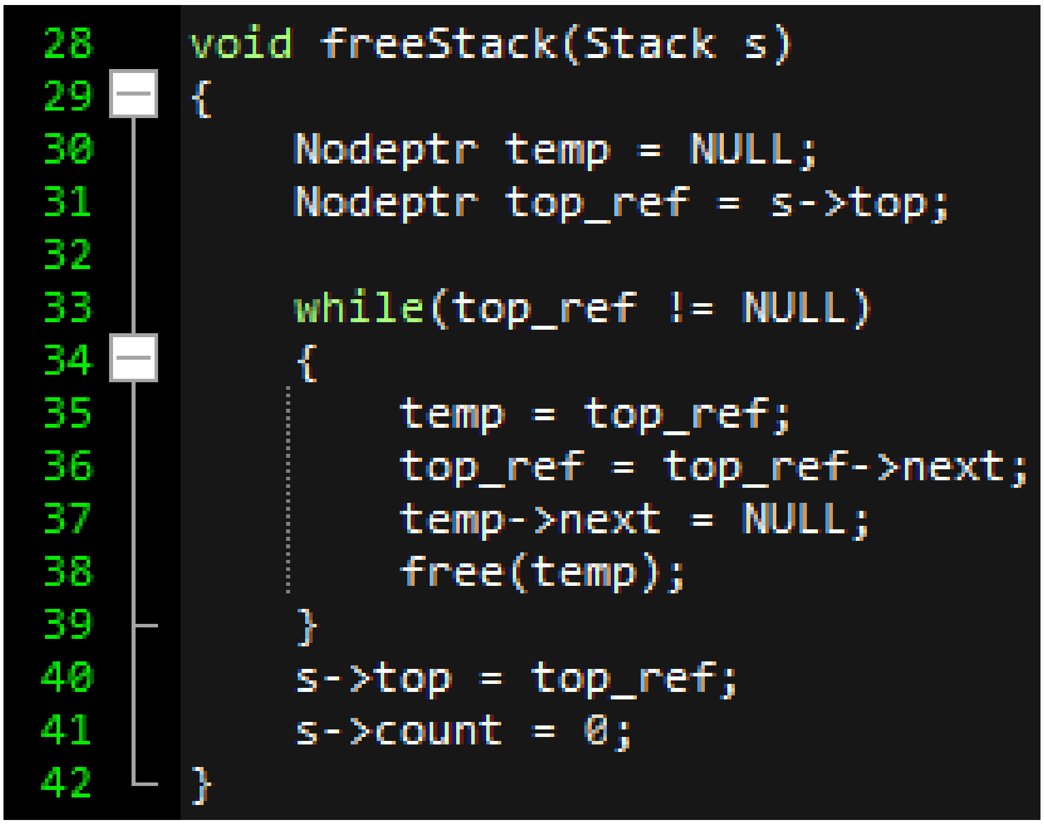


```c
typedef char stackItem;
typedef struct node* Nodeptr;

typedef struct
{
    int count;
    Nodeptr top;
}STACK_HEAD;

typedef STACK_HEAD* Stack;

typedef struct node{
    stackItem data;
    Nodeptr next;
}STACK_NODE;

Stack newStack();
void freeStack(Stack s);
void push(Stack s, stackItem item);                   //Insert at the top
void pop(Stack s);                                    //Deleting the top element
stackItem stackTop(Stack s);                          //returns the top item
int isEmpty(Stack s);
void display(Stack s);
int parenthesesPair(char opening, char closing);     //Checks paired symbols
```

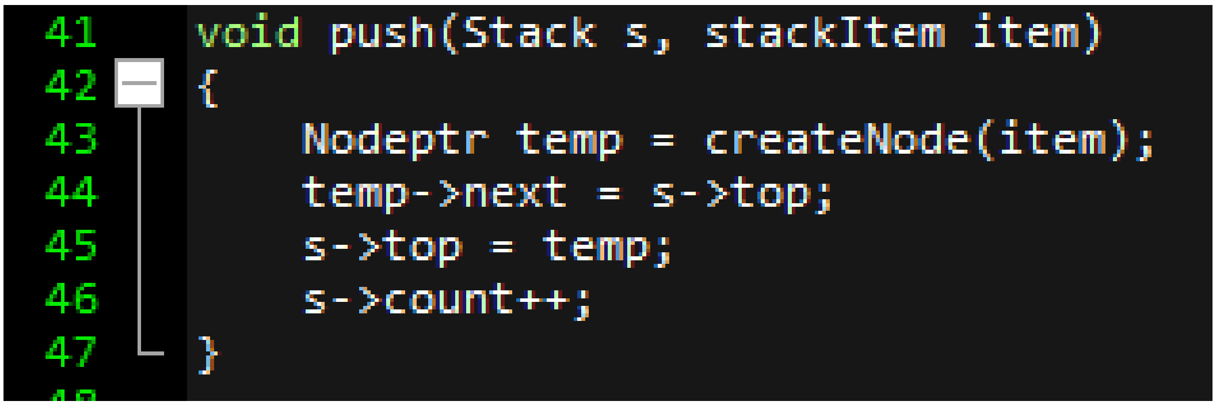
Figure 2. Stack.h



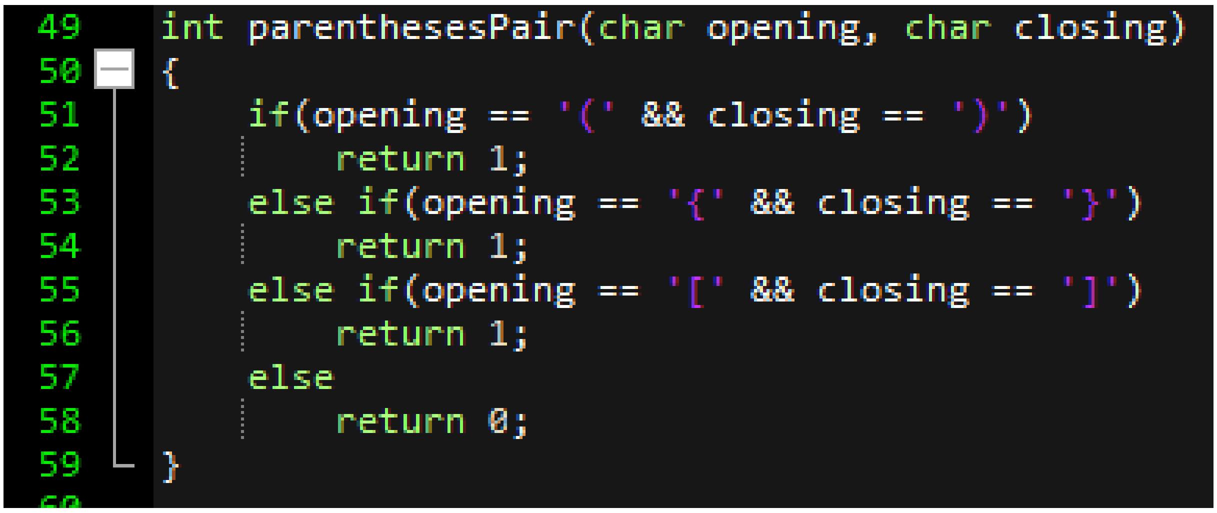
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Stack.h"

Stack newStack()
{
    Stack s;

    s = (Stack)malloc(sizeof(STACK_HEAD));
    s->count = 0;
    s->top = NULL;

    return;
}

Nodeptr createNode(char item)
{
    Nodeptr ptr;

    ptr = (Nodeptr)malloc(sizeof(STACK_NODE));
    ptr->data = item;
    ptr->next = NULL;

    return ptr;
}
```



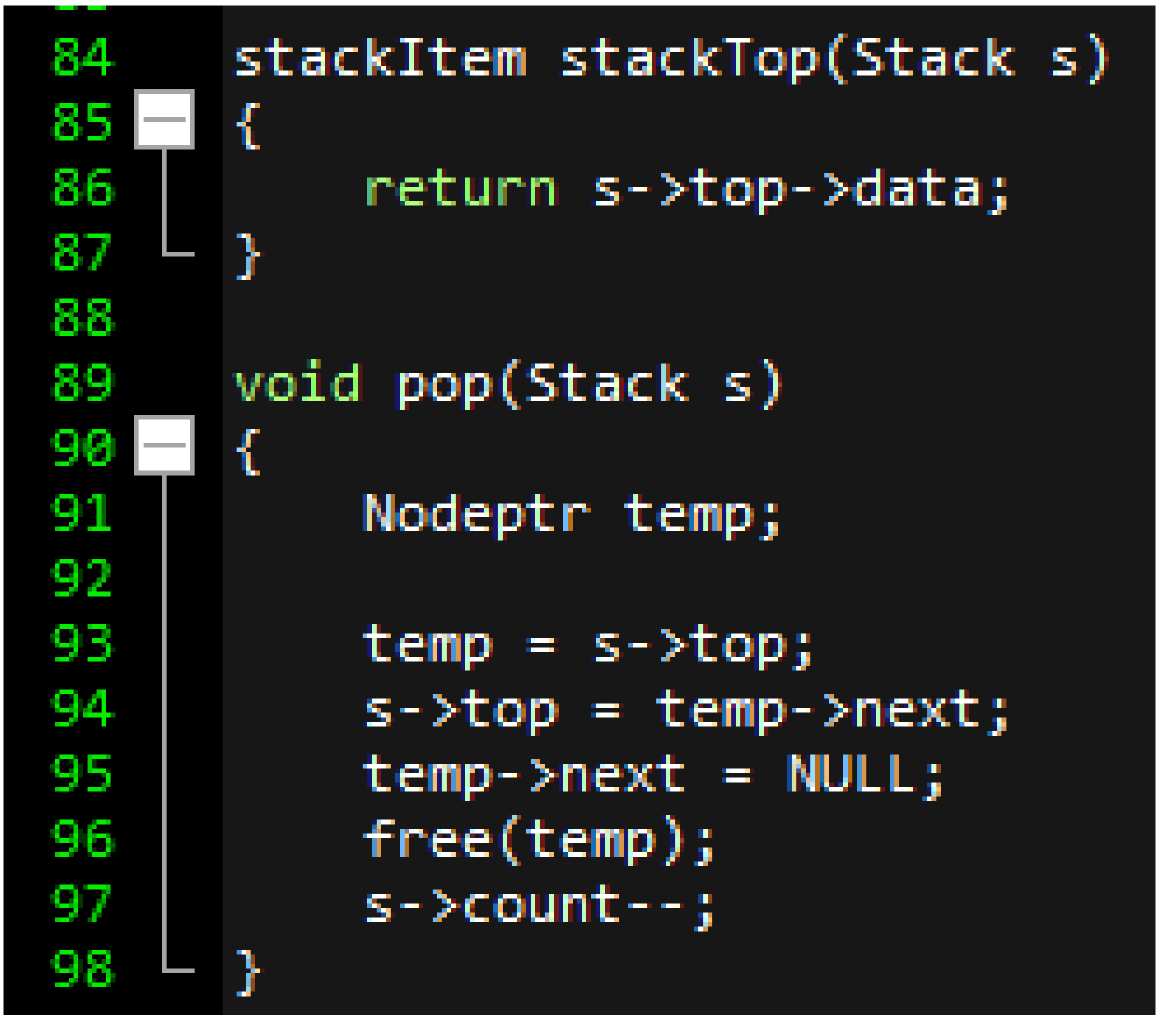
```c
void freeStack(Stack s)
{
    Nodeptr temp = NULL;
    Nodeptr top_ref = s->top;

    while(top_ref != NULL)
    {
        temp = top_ref;
        top_ref = top_ref->next;
        temp->next = NULL;
        free(temp);
    }
    s->top = top_ref;
    s->count = 0;
}
```



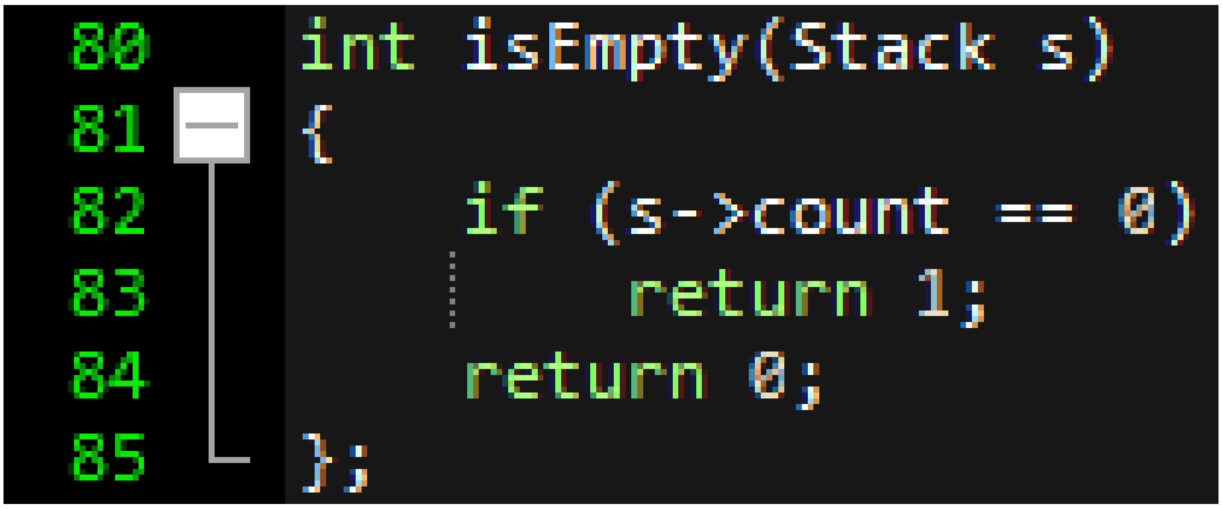
```c
void push(Stack s, stackItem item)
{
    Nodeptr temp = createNode(item);
    temp->next = s->top;
    s->top = temp;
    s->count++;
}
```



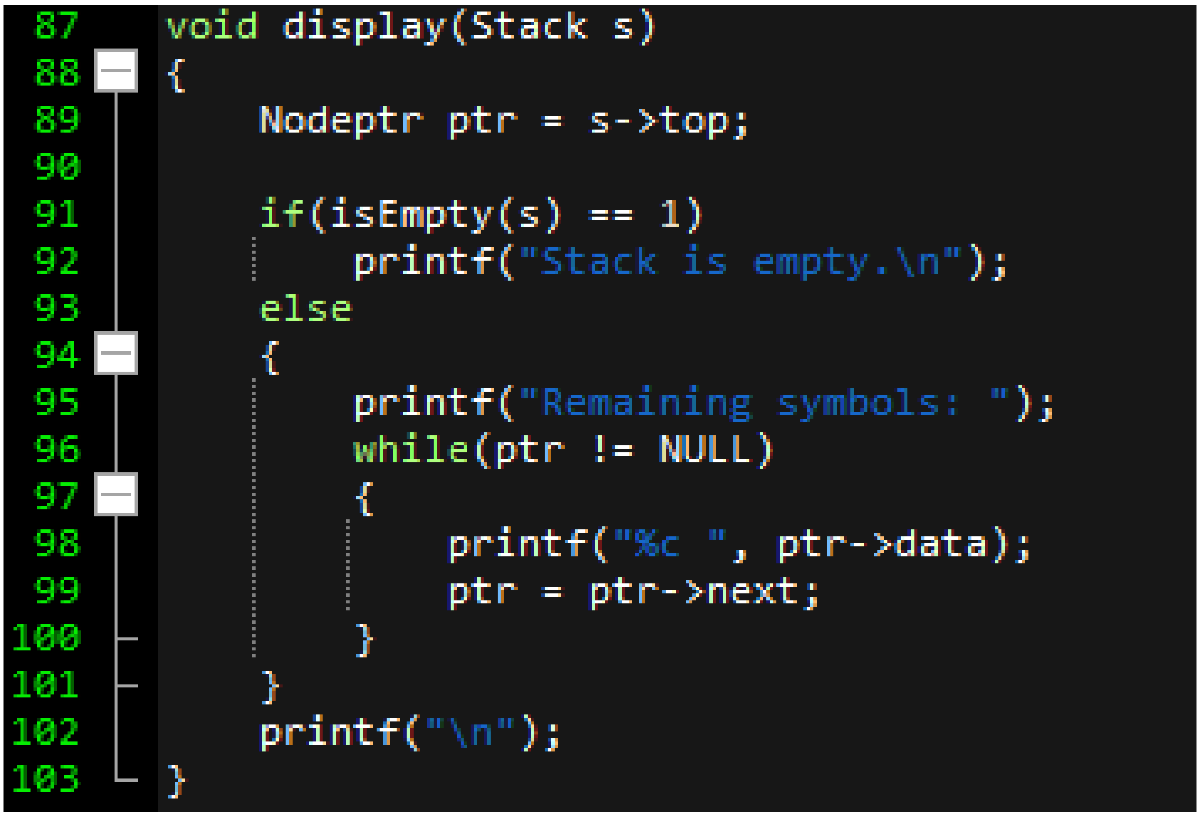
```c
int parenthesesPair(char opening, char closing)
{
    if(opening == '(' && closing == ')')
        return 1;
    else if(opening == '{' && closing == '}')
        return 1;
    else if(opening == '[' && closing == ']')
        return 1;
    else
        return 0;
}
```



```c
stackItem stackTop(Stack s)
{
    return s->top->data;
}

void pop(Stack s)
{
    Nodeptr temp;

    temp = s->top;
    s->top = temp->next;
    temp->next = NULL;
    free(temp);
    s->count--;
}
```

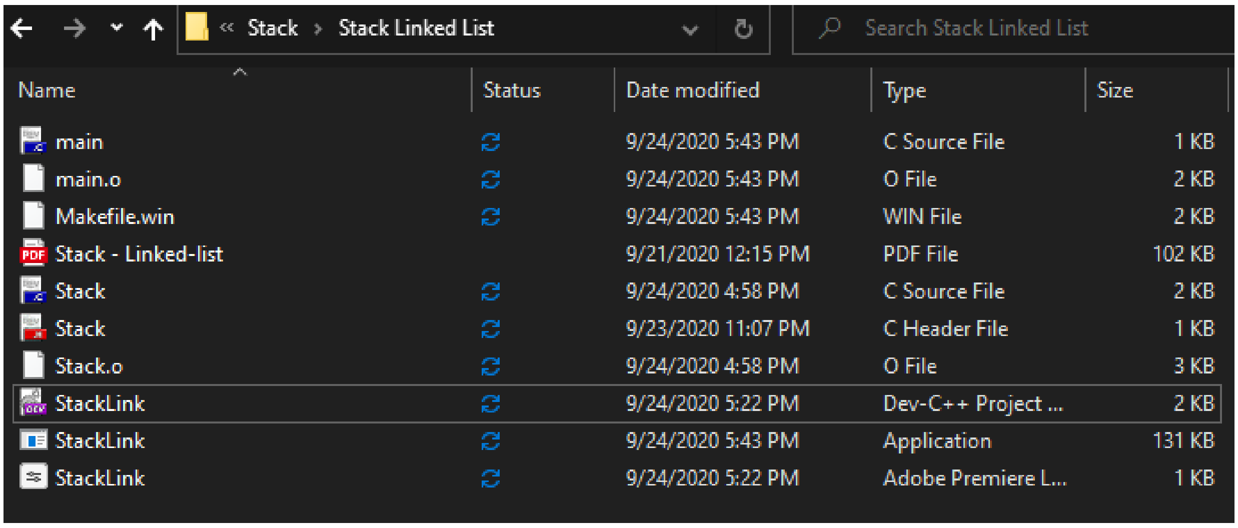


```c
int isEmpty(Stack s)
{
	if (s->count == 0)
		return 1;
	return 0;
};
```



```c
void display(Stack s)
{
	Nodeptr ptr = s->top;
	
	if(isEmpty(s) == 1)
		printf("Stack is empty.\n");
	else
	{
		printf("Remaining symbols: ");
		while(ptr != NULL)
		{
			printf("%c ", ptr->data);
			ptr = ptr->next;
		}
	}
	printf("\n");
}
```

Figure 3. Stack.c



« Stack › Stack Linked List

Search Stack Linked List

Name	Status	Date modified	Type	Size
main		9/24/2020 5:43 PM	C Source File	1 KB
main.o		9/24/2020 5:43 PM	O File	2 KB
Makefile.win		9/24/2020 5:43 PM	WIN File	2 KB
Stack - Linked-list		9/21/2020 12:15 PM	PDF File	102 KB
Stack		9/24/2020 4:58 PM	C Source File	2 KB
Stack		9/23/2020 11:07 PM	C Header File	1 KB
Stack.o		9/24/2020 4:58 PM	O File	3 KB
StackLink		9/24/2020 5:22 PM	Dev-C++ Project ...	2 KB
StackLink		9/24/2020 5:43 PM	Application	131 KB
StackLink		9/24/2020 5:22 PM	Adobe Premiere L...	1 KB

Figure 4. Directory of the Files