DATE:-20/09/22

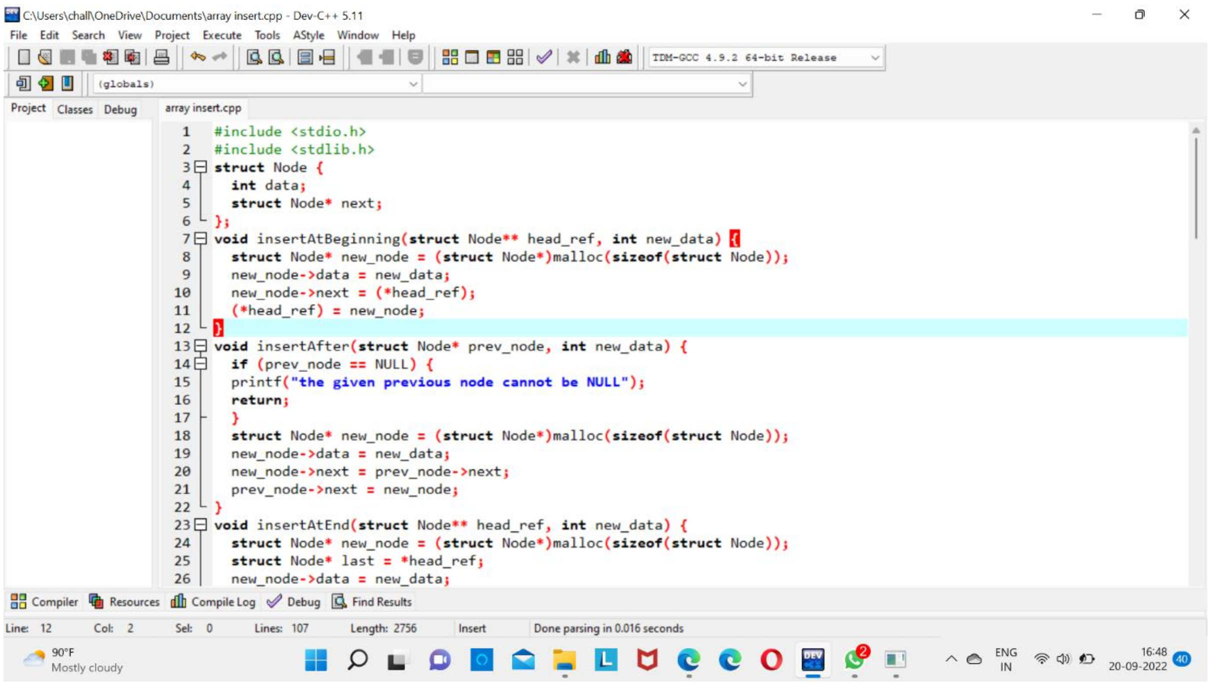
COURSE NAME:-DATA STRUCTURES FOR EXPRESSION EVALUATION

COURSE CODE:-CSA0374

NAME OF THE STUDENT:-CH.INDHU PRIYA

REGNO:-192111191

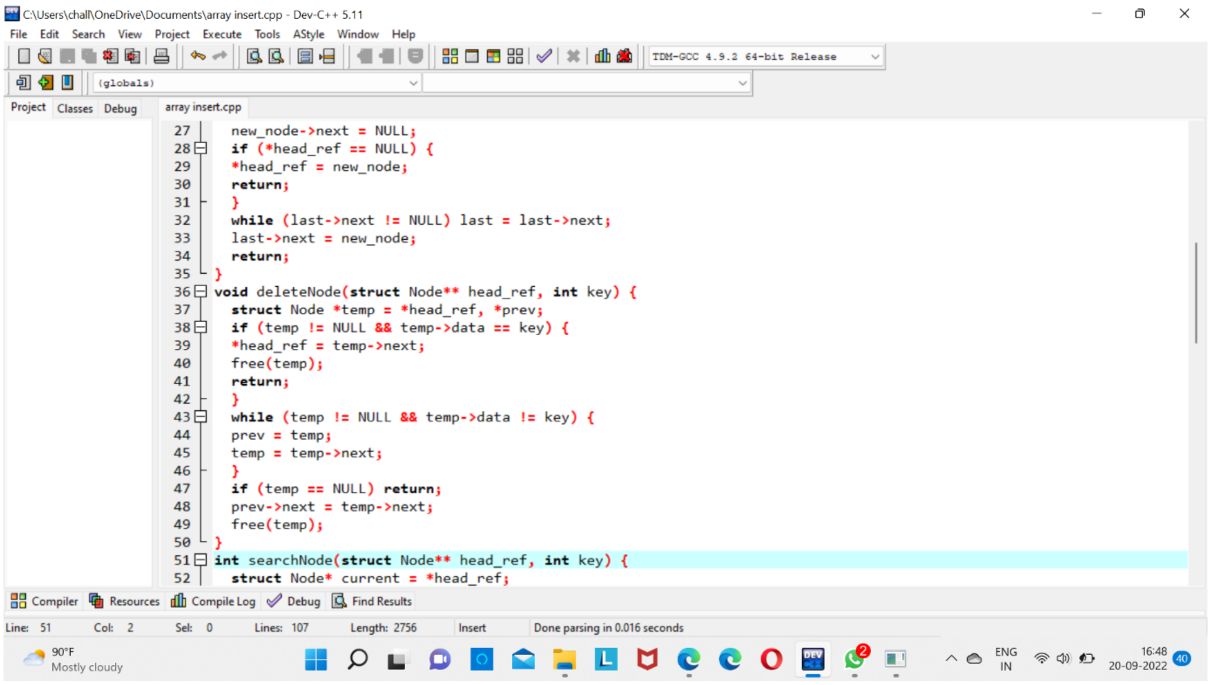
EXPERIMENT:-10(LINKED LIST USING OPERATIONS)


C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
array insert.cpp
Compiler Resources Compile Log Debug Find Results

```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
void insertAtBeginning(struct Node** head_ref, int new_data) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}
void insertAfter(struct Node* prev_node, int new_data) {
    if (prev_node == NULL) {
    printf("the given previous node cannot be NULL");
    return;
    }
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = prev_node->next;
    prev_node->next = new_node;
}
void insertAtEnd(struct Node** head_ref, int new_data) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    struct Node* last = *head_ref;
    new_node->data = new_data;
```

Line: 12 Col: 2 Sel: 0 Lines: 107 Length: 2756 Insert Done parsing in 0.016 seconds

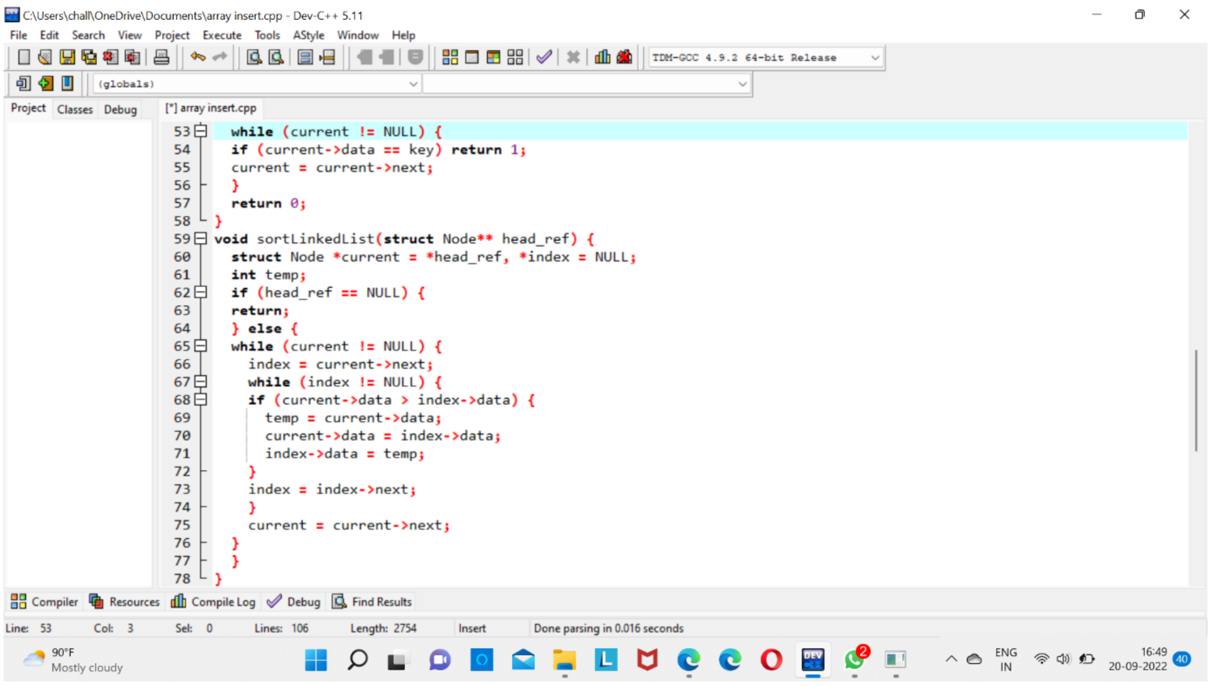
90°F Mostly cloudy ENG IN 16:48 20-09-2022


C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
array insert.cpp
Compiler Resources Compile Log Debug Find Results

```c
    new_node->next = NULL;
    if (*head_ref == NULL) {
    *head_ref = new_node;
    return;
    }
    while (last->next != NULL) last = last->next;
    last->next = new_node;
    return;
}
void deleteNode(struct Node** head_ref, int key) {
    struct Node *temp = *head_ref, *prev;
    if (temp != NULL && temp->data == key) {
    *head_ref = temp->next;
    free(temp);
    return;
    }
    while (temp != NULL && temp->data != key) {
    prev = temp;
    temp = temp->next;
    }
    if (temp == NULL) return;
    prev->next = temp->next;
    free(temp);
}
int searchNode(struct Node** head_ref, int key) {
    struct Node* current = *head_ref;
```

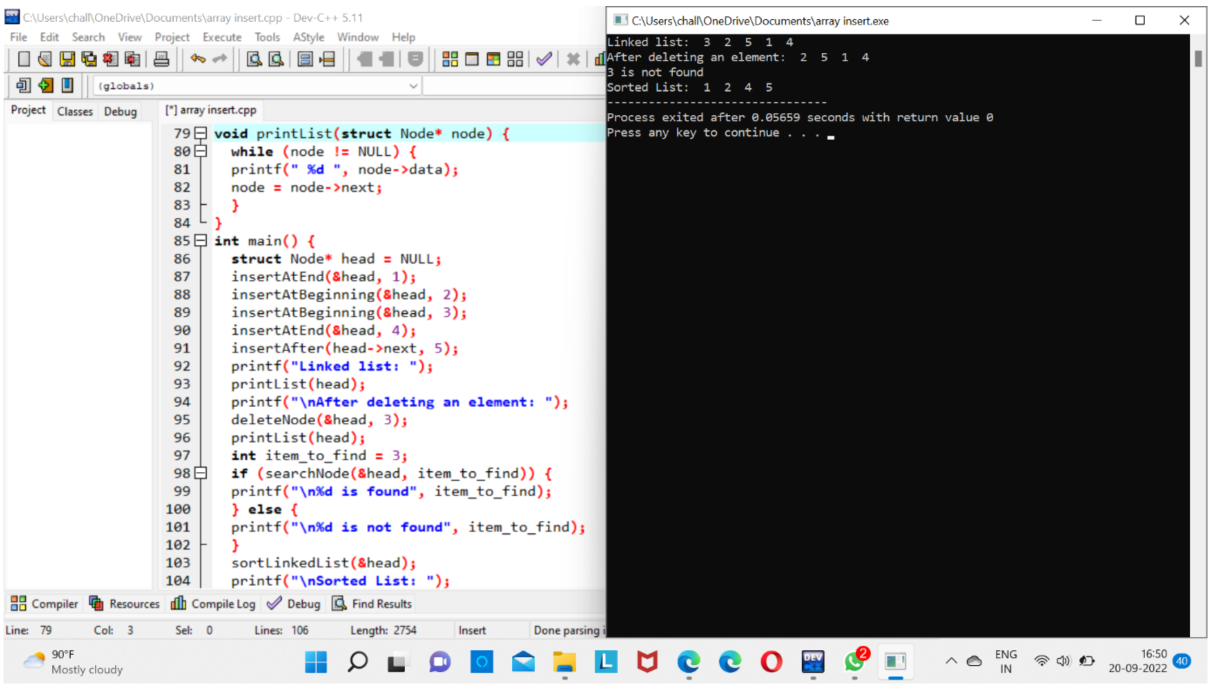
Line: 51 Col: 2 Sel: 0 Lines: 107 Length: 2756 Insert Done parsing in 0.016 seconds

90°F Mostly cloudy ENG IN 16:48 20-09-2022


C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] array insert.cpp
Compiler Resources Compile Log Debug Find Results
90°F
Mostly cloudy
ENG
IN
16:49
20-09-2022

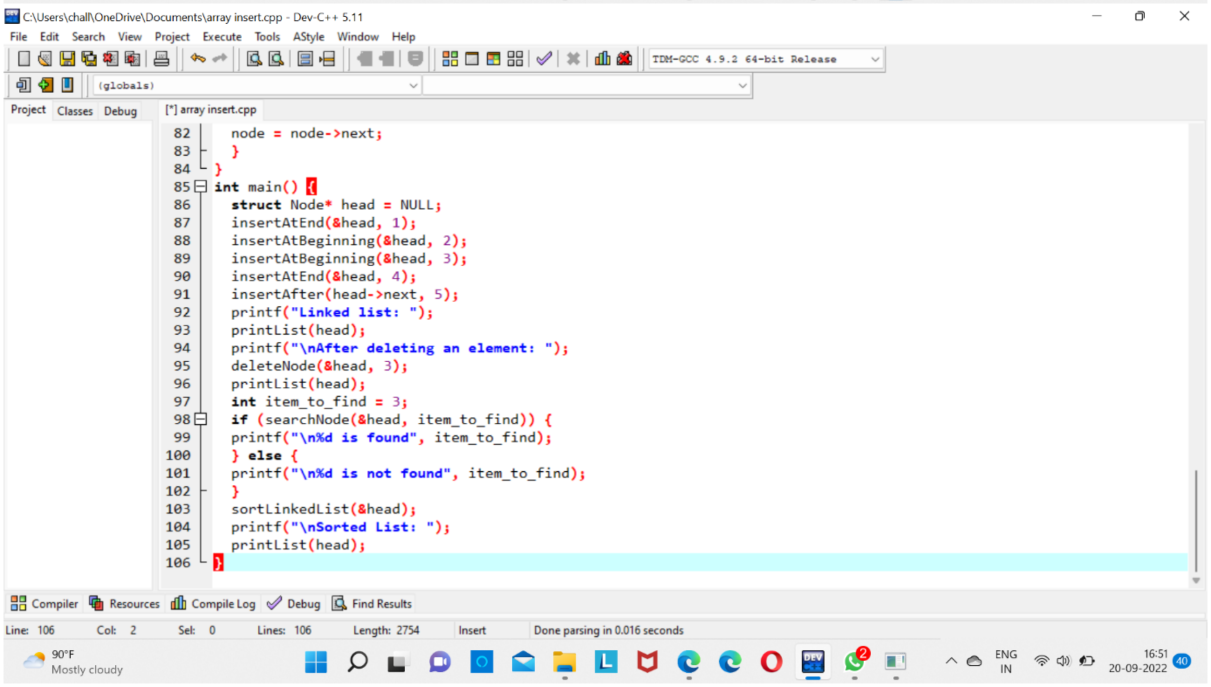
```
 while (current != NULL) {
 if (current->data == key) return 1;
 current = current->next;
 }
 return 0;
}
void sortLinkedList(struct Node** head_ref) {
 struct Node *current = *head_ref, *index = NULL;
 int temp;
 if (head_ref == NULL) {
 return;
 } else {
 while (current != NULL) {
   index = current->next;
   while (index != NULL) {
   if (current->data > index->data) {
     temp = current->data;
     current->data = index->data;
     index->data = temp;
   }
   index = index->next;
   }
   current = current->next;
 }
 }
}
```

```
Line: 53   Col: 3   Sel: 0   Lines: 106   Length: 2754   Insert   Done parsing in 0.016 seconds
```


C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
[*] array insert.cpp
C:\Users\chall\OneDrive\Documents\array insert.exe

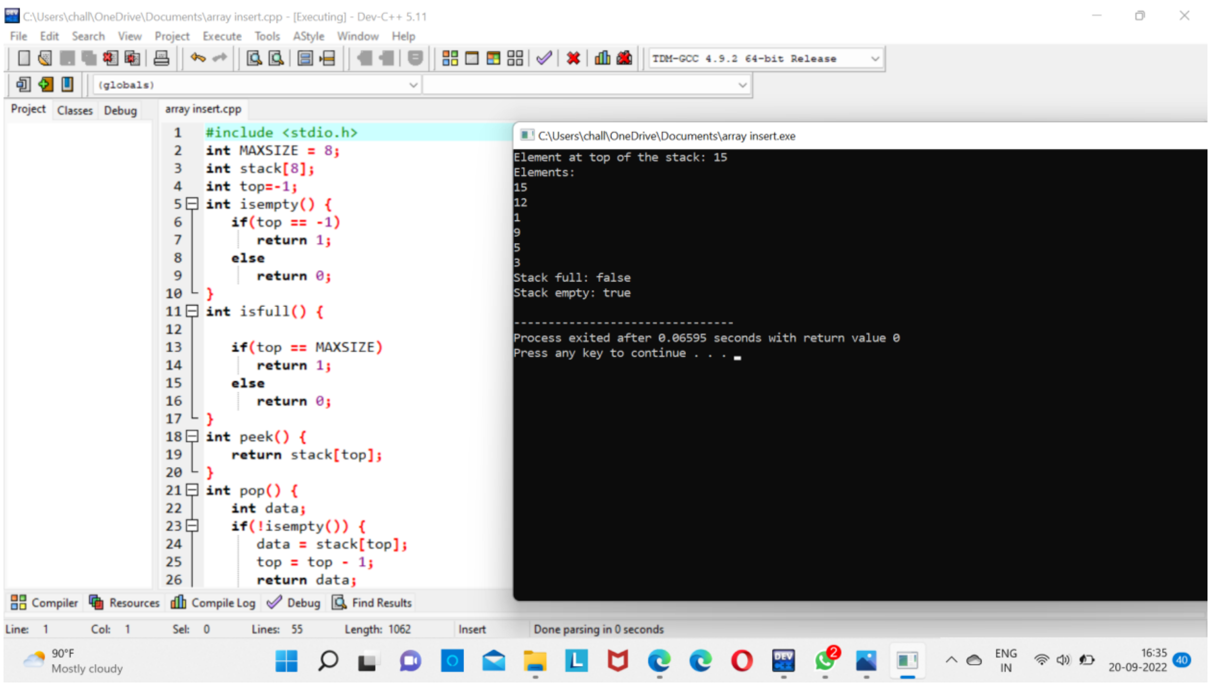
```c
void printList(struct Node* node) {
    while (node != NULL) {
    printf(" %d ", node->data);
    node = node->next;
    }
}
int main() {
    struct Node* head = NULL;
    insertAtEnd(&head, 1);
    insertAtBeginning(&head, 2);
    insertAtBeginning(&head, 3);
    insertAtEnd(&head, 4);
    insertAfter(head->next, 5);
    printf("Linked list: ");
    printList(head);
    printf("\nAfter deleting an element: ");
    deleteNode(&head, 3);
    printList(head);
    int item_to_find = 3;
    if (searchNode(&head, item_to_find)) {
    printf("\n%d is found", item_to_find);
    } else {
    printf("\n%d is not found", item_to_find);
    }
    sortLinkedList(&head);
    printf("\nSorted List: ");
```

```
Linked list:  3  2  5  1  4
After deleting an element:  2  5  1  4
3 is not found
Sorted List:  1  2  4  5
--------------------------------
Process exited after 0.05659 seconds with return value 0
Press any key to continue . . .
```


C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
[*] array insert.cpp
TDM-GCC 4.9.2 64-bit Release

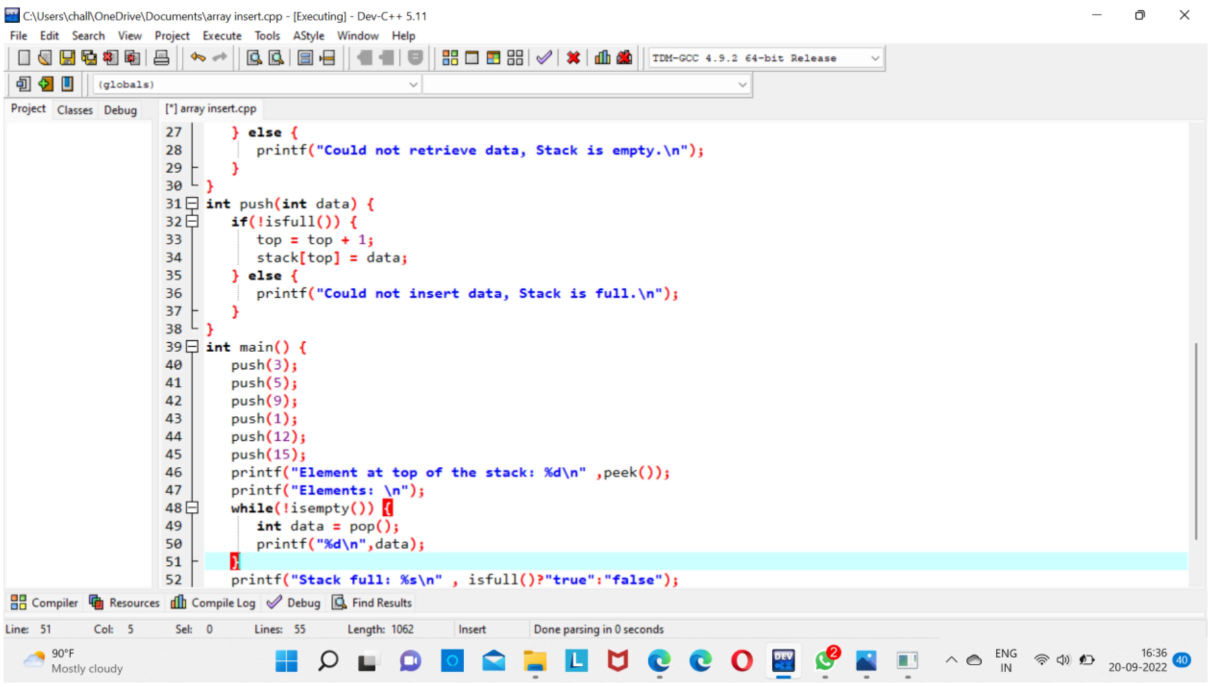
```c
    node = node->next;
    }
}
int main() {
    struct Node* head = NULL;
    insertAtEnd(&head, 1);
    insertAtBeginning(&head, 2);
    insertAtBeginning(&head, 3);
    insertAtEnd(&head, 4);
    insertAfter(head->next, 5);
    printf("Linked list: ");
    printList(head);
    printf("\nAfter deleting an element: ");
    deleteNode(&head, 3);
    printList(head);
    int item_to_find = 3;
    if (searchNode(&head, item_to_find)) {
    printf("\n%d is found", item_to_find);
    } else {
    printf("\n%d is not found", item_to_find);
    }
    sortLinkedList(&head);
    printf("\nSorted List: ");
    printList(head);
}
```

EXPERIMENT:11(STACK OPERATIONS)

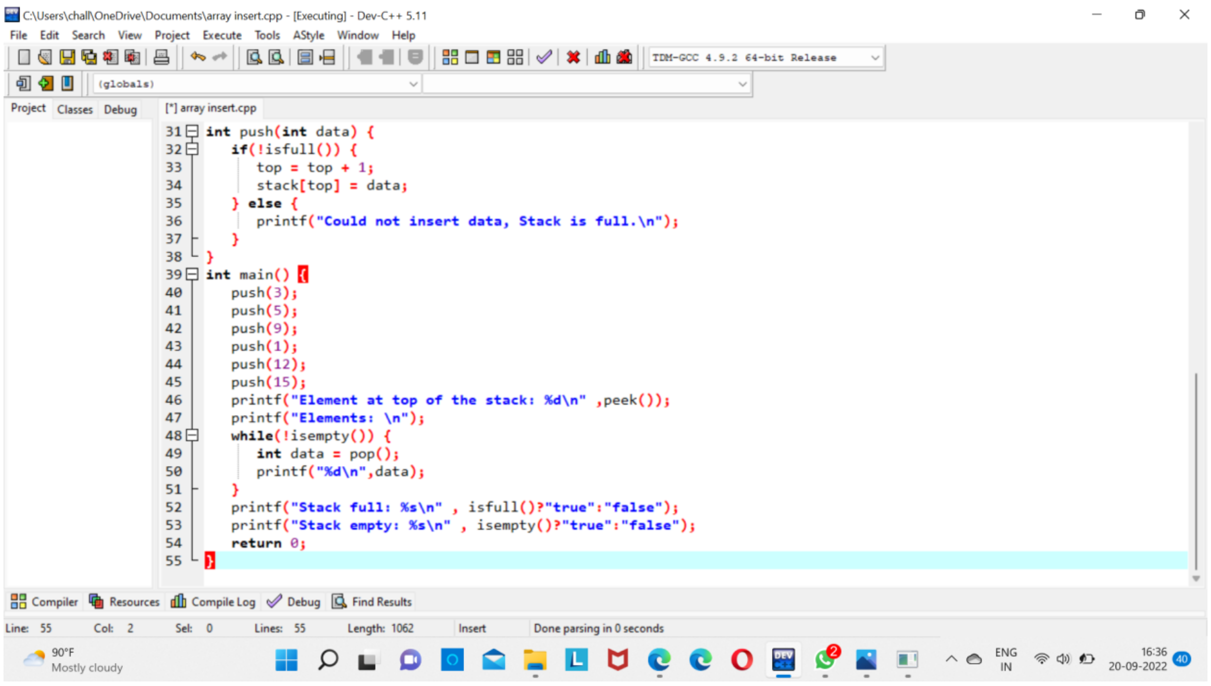

C:\Users\chall\OneDrive\Documents\array insert.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
array insert.cpp
C:\Users\chall\OneDrive\Documents\array insert.exe
Compiler Resources Compile Log Debug Find Results
Line: 1 Col: 1 Sel: 0 Lines: 55 Length: 1062 Insert Done parsing in 0 seconds
90°F Mostly cloudy
ENG IN 16:35 20-09-2022

```
#include <stdio.h>
int MAXSIZE = 8;
int stack[8];
int top=-1;
int isempty() {
   if(top == -1)
      return 1;
   else
      return 0;
}
int isfull() {

   if(top == MAXSIZE)
      return 1;
   else
      return 0;
}
int peek() {
   return stack[top];
}
int pop() {
   int data;
   if(!isempty()) {
      data = stack[top];
      top = top - 1;
      return data;
```

```
Element at top of the stack: 15
Elements:
15
12
1
9
5
3
Stack full: false
Stack empty: true

--------------------------------
Process exited after 0.06595 seconds with return value 0
Press any key to continue . . .
```


C:\Users\chall\OneDrive\Documents\array insert.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] array insert.cpp
Compiler Resources Compile Log Debug Find Results
Line: 51 Col: 5 Sel: 0 Lines: 55 Length: 1062 Insert Done parsing in 0 seconds
90°F Mostly cloudy
ENG IN 16:36 20-09-2022

```
   } else {
      printf("Could not retrieve data, Stack is empty.\n");
   }
}
int push(int data) {
   if(!isfull()) {
      top = top + 1;
      stack[top] = data;
   } else {
      printf("Could not insert data, Stack is full.\n");
   }
}
int main() {
   push(3);
   push(5);
   push(9);
   push(1);
   push(12);
   push(15);
   printf("Element at top of the stack: %d\n" ,peek());
   printf("Elements: \n");
   while(!isempty()) {
      int data = pop();
      printf("%d\n",data);
   }
   printf("Stack full: %s\n" , isfull()?"true":"false");
```


C:\Users\chall\OneDrive\Documents\array insert.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] array insert.cpp
Compiler Resources Compile Log Debug Find Results
Line: 55 Col: 2 Sel: 0 Lines: 55 Length: 1062 Insert Done parsing in 0 seconds
90°F Mostly cloudy
ENG IN
16:36 20-09-2022

```
int push(int data) {
    if(!isfull()) {
        top = top + 1;
        stack[top] = data;
    } else {
        printf("Could not insert data, Stack is full.\n");
    }
}
int main() {
    push(3);
    push(5);
    push(9);
    push(1);
    push(12);
    push(15);
    printf("Element at top of the stack: %d\n" ,peek());
    printf("Elements: \n");
    while(!isempty()) {
        int data = pop();
        printf("%d\n",data);
    }
    printf("Stack full: %s\n" , isfull()?"true":"false");
    printf("Stack empty: %s\n" , isempty()?"true":"false");
    return 0;
}
```