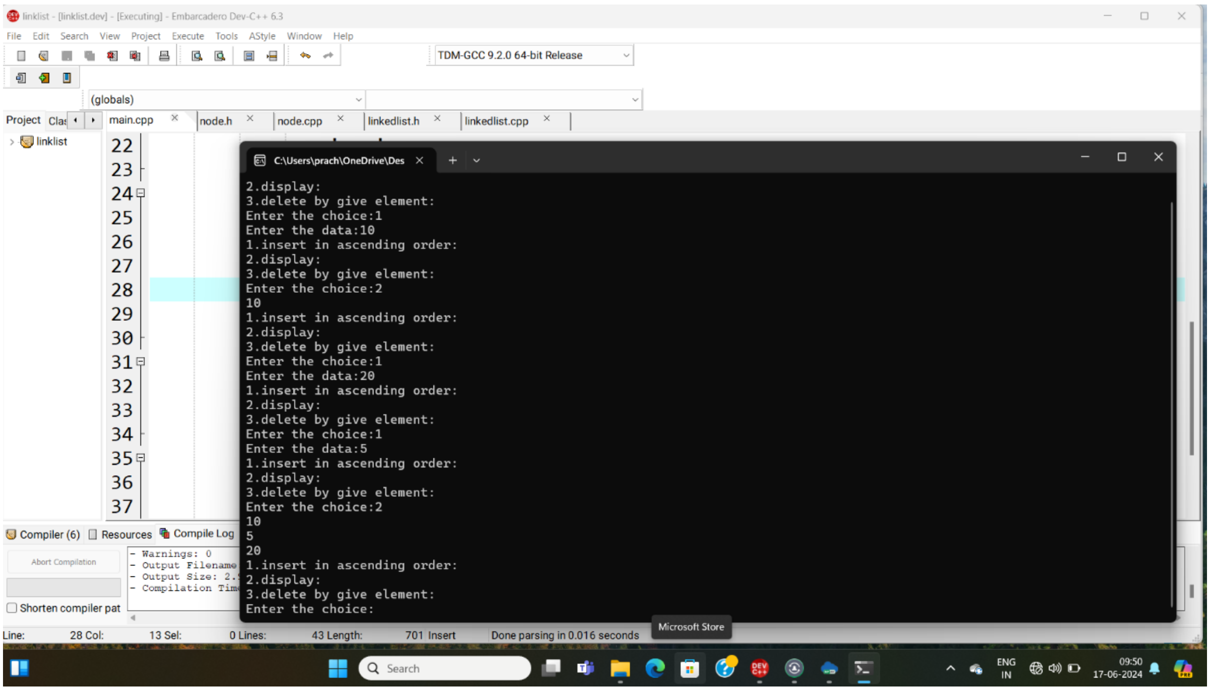
Dsa test

```
1. #include<iostream>
using namespace std;
class Node{
        int data;
        Node* next;

        public:
        Node(int);
        void setData(int);
        void setNext(Node*);
        int getData();
        Node* getNext();
};
#include"node.h"

Node::Node(int data){
        this->data=data;
        this->next=NULL;
}
void Node::setData(int data){
        this->data=data;
}
void Node::setNext(Node* n){
        this->next=n;
}
int Node::getData(){
        return this->data;
}
Node* Node::getNext(){
        return this->next;
```

```
}
#include"node.h"
class linkedList{
        Node* start;
        public:
        linkedList();
        void insertBySort(int);
        void display();
        Node* searchForDelete(int);
        void deleteBYdata(int);
};
#include"linkedlist.h"
linkedList::linkedList(){
        this->start=NULL;
}
void linkedList::insertBySort(int data){
        Node* temp=new Node(data);
        Node* p=start;

        if(start==NULL){
                start=temp;
                return;
        }
        if(p->getNext()==NULL){
                if(temp->getData()<p->getData()){

                        temp->setNext(p);
                        start=temp;
                        return;
                }
                else{
                    p->setNext(temp);
                    return;
                }
        }

        while(p->getNext()->getNext()!=NULL){
                if(temp->getData()<p->getNext()->getData()){
                    temp->setNext(p->getNext());
                    p->setNext(temp);
                    return;
                }
                else{
                    p=p->getNext();
                }
        }
        //for last element
        if(temp->getData()<p->getNext()->getData()){
                    temp->setNext(p->getNext());
                    p->setNext(temp);
                    return;
                }


}
void linkedList::display(){
        Node* p=start;
        while(p!=NULL){

                p=p->getNext();
        }
}
```

```
Node* linkedList::searchForDelete(int data){
        Node* p=start;
        if(p->getNext()==NULL){

                if(p->getData()==data){
                        return p;
                }
        }

        //for 1st element
        if(p->getData()==data){
                cout<<p->getData();
                return p;
        }
        while(p->getNext()->getNext()!=NULL){
                if(p->getNext()->getData()==data){

                        return p;
                }
                else{
                        p=p->getNext();
                }
        }

        //for last element;
        if(p->getNext()->getData()==data){
                return p;
        }

        return NULL;
}

void linkedList::deleteBYdata(int data){
        Node* preAdd=searchForDelete(data);
        Node* q=NULL;
        if(preAdd->getNext()==NULL){
                q=preAdd;
                start=q->getNext();
        }
        if(preAdd!=NULL){
                q=preAdd->getNext();
                cout<<q->getData()<<"is delete";
                preAdd->setNext(q);
                delete q;
                return;
        }
        else{
                cout<<"element not found";
                return;
        }


}
```



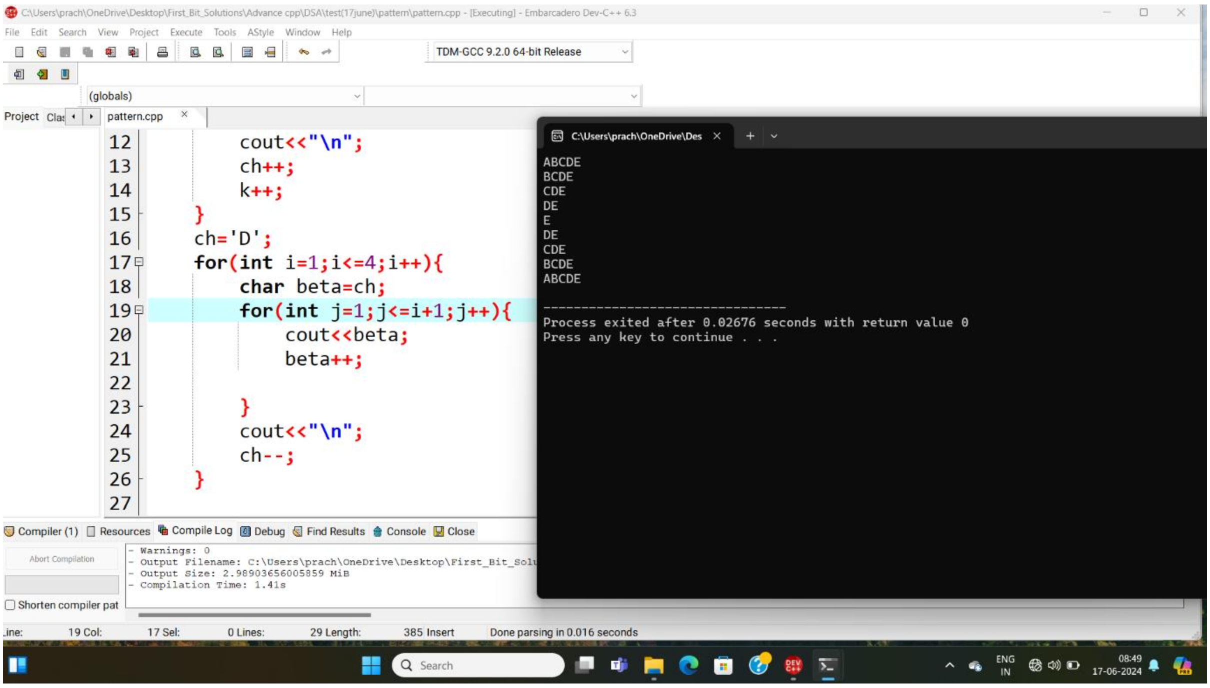
```
- Warnings: 0
- Output Filename
- Output Size: 2.
- Compilation Tim
```

```
2.display:
3.delete by give element:
Enter the choice:1
Enter the data:10
1.insert in ascending order:
2.display:
3.delete by give element:
Enter the choice:2
10
1.insert in ascending order:
2.display:
3.delete by give element:
Enter the choice:1
Enter the data:20
1.insert in ascending order:
2.display:
3.delete by give element:
Enter the choice:1
Enter the data:5
1.insert in ascending order:
2.display:
3.delete by give element:
Enter the choice:2
10
5
20
1.insert in ascending order:
2.display:
3.delete by give element:
Enter the choice:
```

Line: 28 Col: 13 Sel: 0 Lines: 43 Length: 701 Insert Done parsing in 0.016 seconds

2.

```
#include<iostream>
using namespace std;
int main(){
        int k=0;
        char ch='A';
        for(int i=1;i<=5;i++){
                char alpha=ch;
                for(int j=1;j<=5-k;j++){
                        cout<<alpha;
                        alpha++;
                }
                cout<<"\n";
                ch++;
                k++;
        }
        ch='D';
        for(int i=1;i<=4;i++){
                char beta=ch;
                for(int j=1;j<=i+1;j++){
                        cout<<beta;
                        beta++;

                }
                cout<<"\n";
                ch--;
        }

        return 0;
}
```


C:\Users\prach\OneDrive\Desktop\First_Bit_Solutions\Advance cpp\DSA\test(17june)\pattern\pattern.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
pattern.cpp
C:\Users\prach\OneDrive\Des
Compiler (1) Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Search
ENG IN
08:49
17-06-2024

```
        cout<<"\n";
        ch++;
        k++;
    }
    ch='D';
    for(int i=1;i<=4;i++){
        char beta=ch;
        for(int j=1;j<=i+1;j++){
            cout<<beta;
            beta++;

        }
        cout<<"\n";
        ch--;
    }
```

```
ABCDE
BCDE
CDE
DE
E
DE
CDE
BCDE
ABCDE

--------------------------------
Process exited after 0.02676 seconds with return value 0
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\prach\OneDrive\Desktop\First_Bit_Solu
- Output Size: 2.98903656005859 MiB
- Compilation Time: 1.41s
```

Line: 19 Col: 17 Sel: 0 Lines: 29 Length: 385 Insert Done parsing in 0.016 seconds