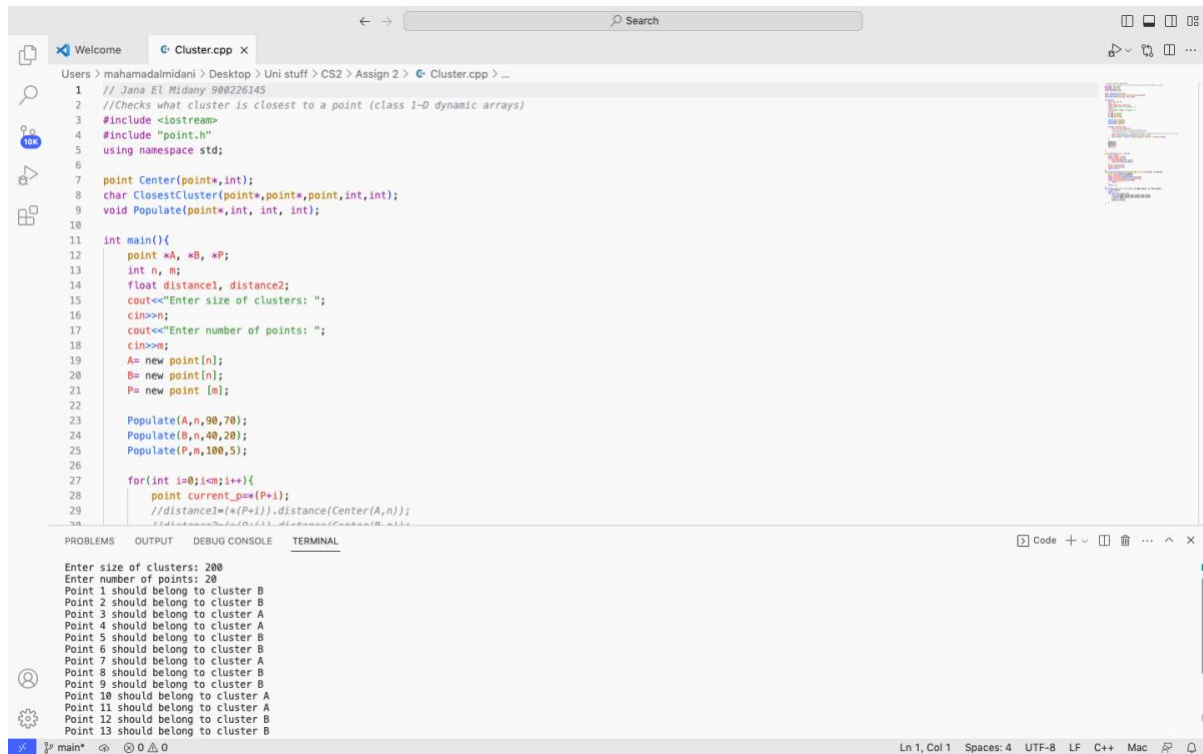
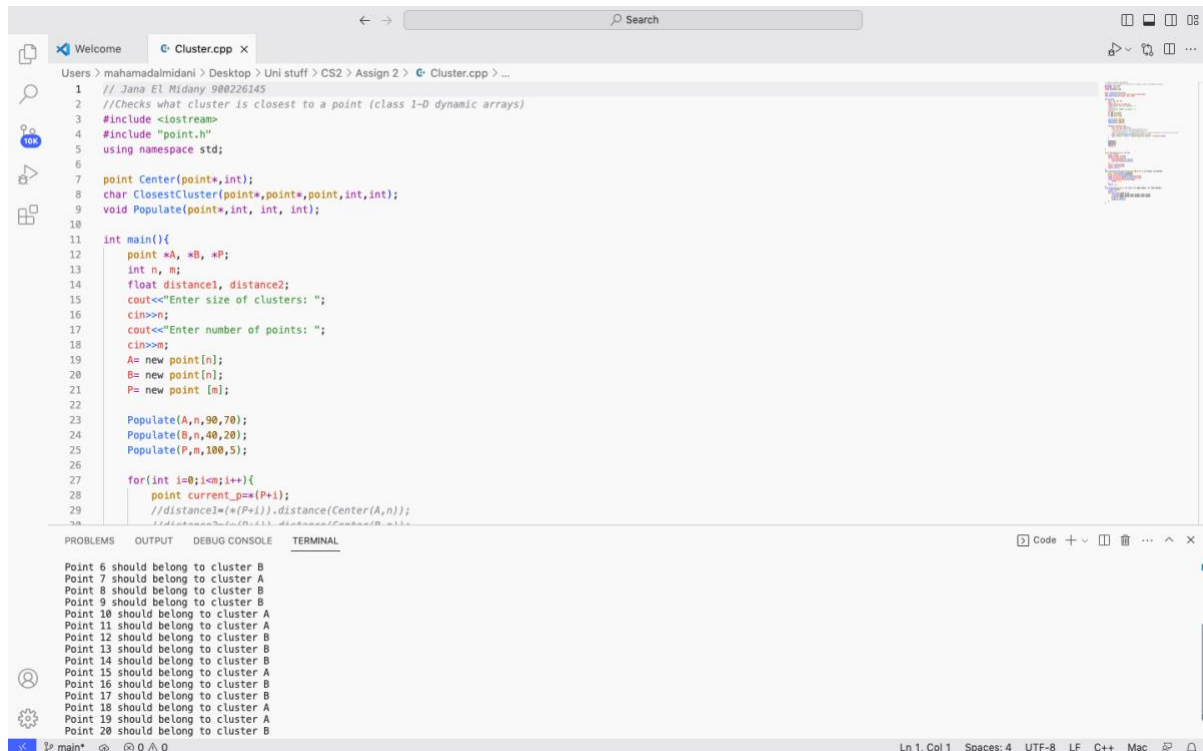


Results for n=200, m=20



```
1 // Jana El Midany 900226145
2 //Checks what cluster is closest to a point (class 1-D dynamic arrays)
3 #include <iostream>
4 #include "point.h"
5 using namespace std;
6
7 point Center(point*,int);
8 char ClosestCluster(point*,point*,point,int,int);
9 void Populate(point*,int, int, int);
10
11 int main(){
12     point *A, *B, *P;
13     int n, m;
14     float distance1, distance2;
15     cout<<"Enter size of clusters: ";
16     cin>>n;
17     cout<<"Enter number of points: ";
18     cin>>m;
19     A= new point[n];
20     B= new point[n];
21     P= new point [m];
22
23     Populate(A,n,90,70);
24     Populate(B,n,40,20);
25     Populate(P,n,100,5);
26
27     for(int i=0;i<m;i++){
28         point current_p=*(P+i);
29         //distance1=*(P+i).distance(Center(A,n));
30         //distance2=*(P+i).distance(Center(B,m));
31     }
32 }
```

Enter size of clusters: 200
Enter number of points: 20
Point 1 should belong to cluster B
Point 2 should belong to cluster B
Point 3 should belong to cluster A
Point 4 should belong to cluster A
Point 5 should belong to cluster B
Point 6 should belong to cluster B
Point 7 should belong to cluster A
Point 8 should belong to cluster B
Point 9 should belong to cluster B
Point 10 should belong to cluster A
Point 11 should belong to cluster A
Point 12 should belong to cluster B
Point 13 should belong to cluster B



```
1 // Jana El Midany 900226145
2 //Checks what cluster is closest to a point (class 1-D dynamic arrays)
3 #include <iostream>
4 #include "point.h"
5 using namespace std;
6
7 point Center(point*,int);
8 char ClosestCluster(point*,point*,point,int,int);
9 void Populate(point*,int, int, int);
10
11 int main(){
12     point *A, *B, *P;
13     int n, m;
14     float distance1, distance2;
15     cout<<"Enter size of clusters: ";
16     cin>>n;
17     cout<<"Enter number of points: ";
18     cin>>m;
19     A= new point[n];
20     B= new point[n];
21     P= new point [m];
22
23     Populate(A,n,90,70);
24     Populate(B,n,40,20);
25     Populate(P,n,100,5);
26
27     for(int i=0;i<m;i++){
28         point current_p=*(P+i);
29         //distance1=*(P+i).distance(Center(A,n));
30         //distance2=*(P+i).distance(Center(B,m));
31     }
32 }
```

Point 6 should belong to cluster B
Point 7 should belong to cluster A
Point 8 should belong to cluster B
Point 9 should belong to cluster B
Point 10 should belong to cluster A
Point 11 should belong to cluster A
Point 12 should belong to cluster B
Point 13 should belong to cluster B
Point 14 should belong to cluster B
Point 15 should belong to cluster A
Point 16 should belong to cluster B
Point 17 should belong to cluster B
Point 18 should belong to cluster A
Point 19 should belong to cluster A
Point 20 should belong to cluster B