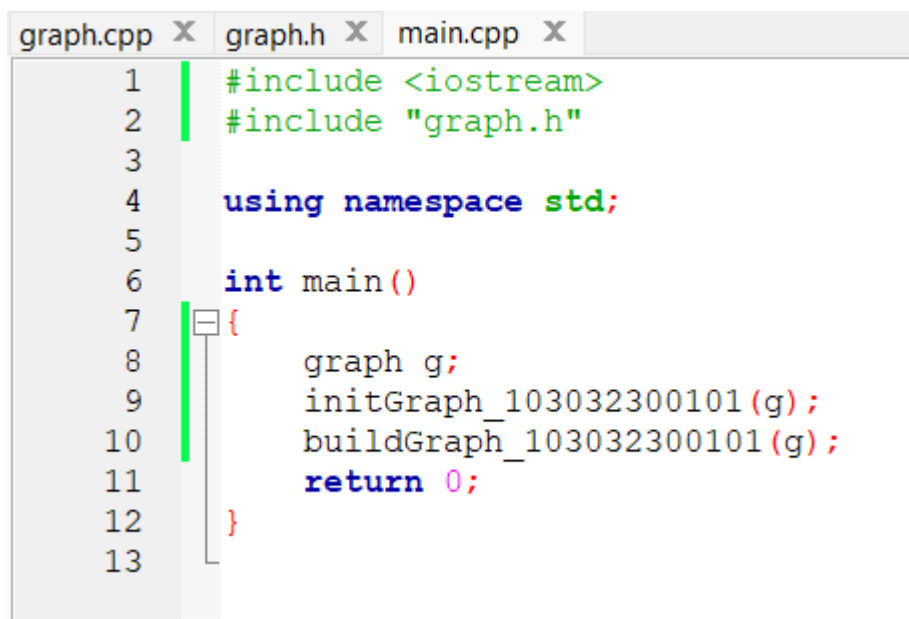


Made Naradeon Handika Pramesta  
103032300101

Main.cpp



```
graph.cpp X graph.h X main.cpp X
1  #include <iostream>
2  #include "graph.h"
3
4  using namespace std;
5
6  int main()
7  {
8      graph g;
9      initGraph_103032300101(g);
10     buildGraph_103032300101(g);
11     return 0;
12 }
13
```

Graph.h

```
graph.cpp X graph.h X main.cpp X
1  #ifndef GRAPH_H_INCLUDED
2  #define GRAPH_H_INCLUDED
3  #include <iostream>
4
5  using namespace std;
6
7  typedef struct vertex* adrVertex;
8  typedef struct edge *adrEdge;
9
10 struct vertex {
11     char idVertex;
12     adrVertex nextVertex;
13     adrEdge firstEdge;
14 };
15
16 struct edge {
17     char destVertex;
18     int weight;
19     adrEdge nextEdge;
20 };
21
22 struct graph {
23     adrVertex firstVertex;
24 };
25
26 void createVertex_103032300101(char newVertexID, adrVertex &v);
27 void initGraph_103032300101(graph &g);
28 void addVertex_103032300101(graph &g, char newVertexID);
29 void buildGraph_103032300101(graph &g);
30
31
32
33 #endif // GRAPH_H_INCLUDED
34
```

Graph.cpp

```
graph.cpp X graph.h X main.cpp X
1  #include "graph.h"
2
3  void createVertex_103032300101(char newVertexID, adrVertex &v) {
4      v->idVertex = newVertexID;
5      v->firstEdge = NULL;
6      v->nextVertex = NULL;
7  }
8
9  void initGraph_103032300101(graph &g) {
10     g.firstVertex = NULL;
11 }
12
13 void addVertex_103032300101(graph &g, char newVertexID) {
14     adrVertex v = new vertex;
15     createVertex_103032300101(newVertexID, v);
16     if (g.firstVertex == NULL) {
17         g.firstVertex = v;
18     } else {
19         adrVertex q = g.firstVertex;
20         while(q->nextVertex != NULL) {
21             q = {adrVertex vertex::nextVertex
22         }
23         q->nextVertex = v;
24     }
25 }
26
27 void buildGraph_103032300101(graph &g) {
28     char input;
29     cin >> input;
30     while (input >= 'A' && input <= 'Z') {
```

```
graph.cpp X graph.h X main.cpp X
24     }
25 }
26
27 void buildGraph_103032300101(graph &g) {
28     char input;
29     cin >> input;
30     while (input >= 'A' && input <= 'Z') {
31         if (g.firstVertex == NULL) {
32             addVertex_103032300101(g, input);
33         } else {
34             bool unique = true;
35             adrVertex v = g.firstVertex;
36             while (v != NULL && unique) {
37                 if (v->idVertex == input) {
38                     unique = false;
39                 }
40                 v = v->nextVertex;
41             }
42             if (unique) {
43                 addVertex_103032300101(g, input);
44             } else {
45                 cout << "ID Sudah ada" << endl;
46             }
47         }
48         cin >> input;
49     }
50 }
51
52
53
```