breadth first search code in c.p | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO >

main.c

Share | Run | Output | Clear

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
} Node;
typedef struct QueueNode {
    Node* node;
    struct QueueNode* next;
} QueueNode;
Node* createNode(int data) {
    Node* newNode = (Node*) malloc(sizeof(Node));
    if (!newNode) {
        printf("Memory error\n");
        return NULL;
    }
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
QueueNode* createQueueNode(Node* node) {
    QueueNode* newQueueNode = (QueueNode*) malloc(sizeof(QueueNode));
    if (!newQueueNode) {
        printf("Memory error\n");
        return NULL;
    }
    newQueueNode->node = node;
    newQueueNode->next = NULL;
    return newQueueNode;
}
```

breadth first search code in c.p | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO >

main.c

Share | Run | Output | Clear

```c
void enqueue(QueueNode** queue, Node* node) {
    QueueNode* newQueueNode = createQueueNode(node);
    if (*queue == NULL) {
        *queue = newQueueNode;
    } else {
        QueueNode* temp = *queue;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newQueueNode;
    }
}
Node* dequeue(QueueNode** queue) {
    if (*queue == NULL) {
        return NULL;
    }
    Node* node = (*queue)->node;
    QueueNode* temp = *queue;
    *queue = (*queue)->next;
    free(temp);
    return node;
}
void bfs(Node* startNode) {
    QueueNode* queue = NULL;
    enqueue(&queue, startNode);
    while (queue != NULL) {
        Node* node = dequeue(&queue);
        printf("%d ", node->data);
        Node* temp = node->next;
        while (temp != NULL) {
```

breadth first search code in c.p | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Paused

Programiz
C Online Compiler

Sponsored by: Do not miss this opportunity
How to start earning a passive income from home for Ages 35-60
Amazon CFO: Your Route to a Second Income

Programiz PRO >

main.c

Share | Run | Output | Clear

```
52  }
53  void bfs(Node* startNode) {
54      QueueNode* queue = NULL;
55      enqueue(&queue, startNode);
56      while (queue != NULL) {
57          Node* node = dequeue(&queue);
58          printf("%d ", node->data);
59          Node* temp = node->next;
60          while (temp != NULL) {
61              enqueue(&queue, temp);
62              temp = temp->next;
63          }
64      }
65  }
66  int main() {
67      Node* node1 = createNode(1);
68      Node* node2 = createNode(2);
69      Node* node3 = createNode(3);
70      Node* node4 = createNode(4);
71      Node* node5 = createNode(5);
72      node1->next = node2;
73      node2->next = node3;
74      node3->next = node4;
75      node4->next = node5;
76      printf("BFS Traversal: ");
77      bfs(node1);
78      printf("\n");
79      return 0;
80  }
```

breadth first search code in c.p | Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

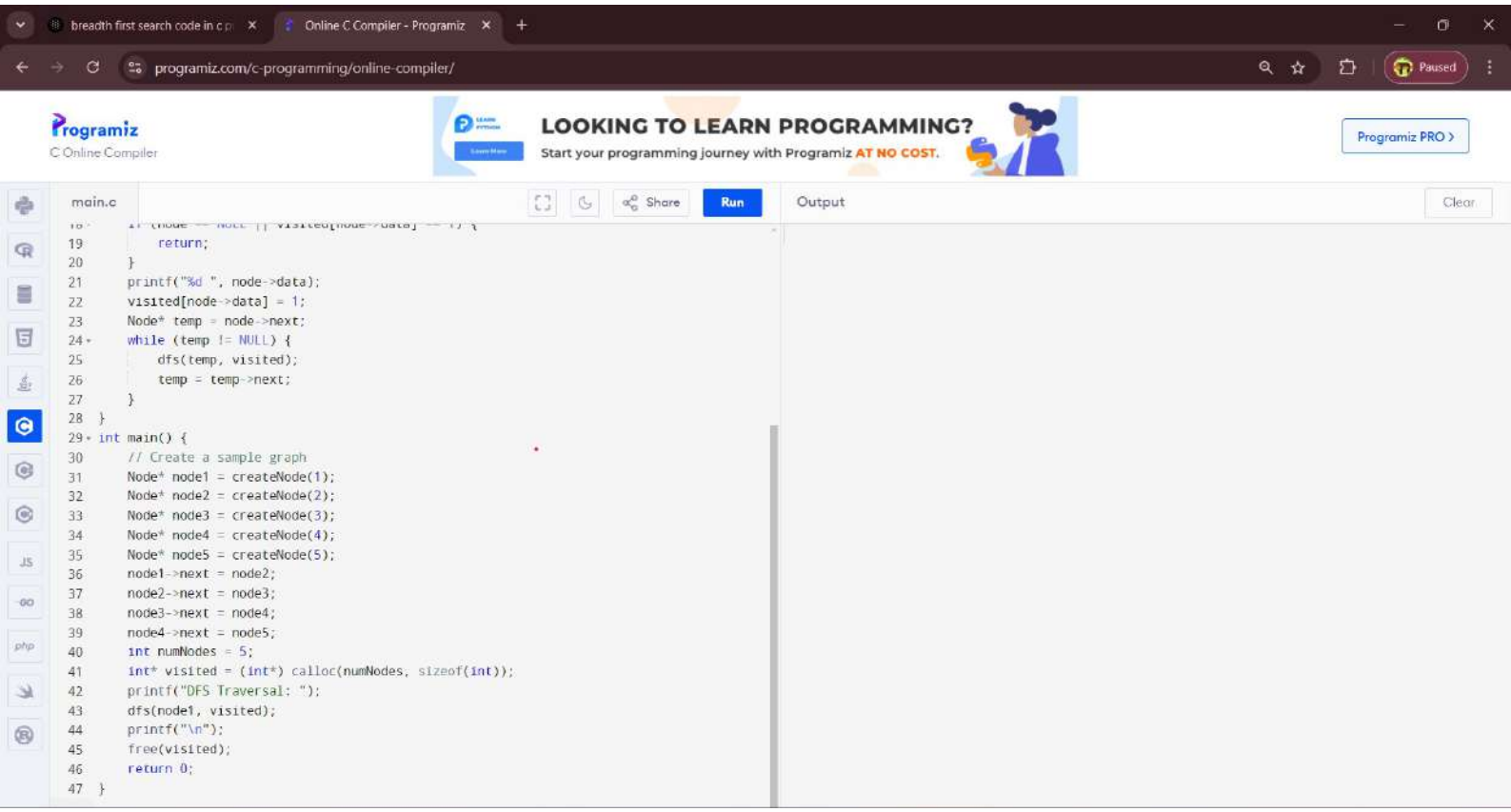
Programiz
C Online Compiler

Programiz PRO >

main.c

Share | Run | Output | Clear

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
} Node;
Node* createNode(int data) {
    Node* newNode = (Node*) malloc(sizeof(Node));
    if (!newNode) {
        printf("Memory error\n");
        return NULL;
    }
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
void dfs(Node* node, int* visited) {
    if (node == NULL || visited[node->data] == 1) {
        return;
    }
    printf("%d ", node->data);
    visited[node->data] = 1;
    Node* temp = node->next;
    while (temp != NULL) {
        dfs(temp, visited);
        temp = temp->next;
    }
}
int main() {
    // Create a sample graph
```


breadth first search code in c.p
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Programiz PRO ›
main.c
Share
Run
Output
Clear
19 return;
20 }
21 printf("%d ", node->data);
22 visited[node->data] = 1;
23 Node* temp = node->next;
24 while (temp != NULL) {
25 dfs(temp, visited);
26 temp = temp->next;
27 }
28 }
29 int main() {
30 // Create a sample graph
31 Node* node1 = createNode(1);
32 Node* node2 = createNode(2);
33 Node* node3 = createNode(3);
34 Node* node4 = createNode(4);
35 Node* node5 = createNode(5);
36 node1->next = node2;
37 node2->next = node3;
38 node3->next = node4;
39 node4->next = node5;
40 int numNodes = 5;
41 int* visited = (int*) calloc(numNodes, sizeof(int));
42 printf("DFS Traversal: ");
43 dfs(node1, visited);
44 printf("\n");
45 free(visited);
46 return 0;
47 }