Online C Compiler - Programiz

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

Paused

All Bookmarks

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;
int main() {
    Node* head = (Node*)malloc(sizeof(Node));
    Node* second = (Node*)malloc(sizeof(Node));
    Node* third = (Node*)malloc(sizeof(Node));
    head->data = 1;
    second->data = 2;
    third->data = 3;
    head->next = second;
    second->next = third;
    third->next = head;
    Node* current = head;
    while (1) {
        printf("%d ", current->data);
        current = current->next;
        if (current == head) break;
    }
    printf("\n");
    return 0;
}
```

```
/tmp/ddsCZcOlwT.o
1 2 3


=== Code Execution Successful ===
```

Online C Compiler - Programiz

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

Paused

All Bookmarks

Programiz
C Online Compiler

main.c

Share

Run

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
} Node;
int main() {
    Node* head = NULL;
    Node* tail = NULL;
    Node* node1 = (Node*) malloc(sizeof(Node));
    node1->data = 10;
    node1->next = NULL;
    node1->prev = NULL;
    head = node1;
    tail = node1;
    Node* node2 = (Node*) malloc(sizeof(Node));
    node2->data = 20;
    node2->next = NULL;
    node2->prev = tail;
    tail->next = node2;
    tail = node2;
    Node* node3 = (Node*) malloc(sizeof(Node));
    node3->data = 30;
    node3->next = NULL;
    node3->prev = tail;
    tail->next = node3;
    tail = node3;
    Node* temp = head;
    while (temp!= NULL) {
```

Output

Clear

```
/tmp/mn5Fj0jI2x.o
10 20 30


=== Code Execution Successful ===
```

Online C Compiler - Programiz

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

Paused

All Bookmarks

Programiz
C Online Compiler

main.c

Share

Run

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node
{
    int data;
    struct Node* next;
} Node;
void insert(Node** head, int data)
{
    Node* newNode = malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = *head;
    *head = newNode;
}
void print(Node* head) {
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
int main() {
    Node* head = NULL;
    insert(&head, 1);
    insert(&head, 2);
    insert(&head, 3);
    print(head);
    return 0;
}
```

Output

Clear

```
/tmp/hV7YMOHJMT.o
3 -> 2 -> 1 -> NULL


=== Code Execution Successful ===
```

Online C Compiler - Programiz

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

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>
struct node {
  int value;
  struct node *next;
};
void printLinkedlist(struct node *p) {
  while (p != NULL) {
    printf("%d ", p->value);
    p = p->next;
  }
}
int main() {
  struct node *head;
  struct node *one = NULL;
  struct node *two = NULL;
  struct node *three = NULL;
  one = malloc(sizeof(struct node));
  two = malloc(sizeof(struct node));
  three = malloc(sizeof(struct node));
  one->value = 1;
  two->value = 2;
  three->value = 3;
  one->next = two;
  two->next = three;
  three->next = NULL;
  head = one;
  printLinkedlist(head);
}
```

Output | Clear

```
/tmp/UcuT5r9wVk.o
1 2 3

=== Code Execution Successful ===
```

Online C Compiler - Programiz

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

Paused

All Bookmarks

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share

Run

```c
#include <stdio.h>
#define MAX_SIZE 5
int stack[MAX_SIZE];
int top = -1;
void push(int element) {
    if (top == MAX_SIZE - 1) {
        printf("Stack is full.\n");
        return;
    }
    stack[++top] = element;
}
int pop() {
    if (top == -1) {
        printf("Stack is empty.\n");
        return -1;
    }
    return stack[top--];
}
int main() {
    push(1);
    push(2);
    push(3);
    printf("Popped: %d\n", pop());
    printf("Popped: %d\n", pop());
    return 0;
}
```

Output

Clear

```
/tmp/TbrLx8vaNH.o
Popped: 3
Popped: 2


=== Code Execution Successful ===S
```

Online C Compiler - Programiz

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

Paused

All Bookmarks

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share

Run

```c
#include <stdio.h>
#include <stdbool.h>
#define MAX_SIZE 100
typedef struct {
    int data[MAX_SIZE];
    int top;
} Stack;
void initialize(Stack *stack) {
    stack->top = -1;
}
bool isFull(Stack *stack) {
    return stack->top == MAX_SIZE - 1;
}
bool isEmpty(Stack *stack) {
    return stack->top == -1;
}
void push(Stack *stack, int element) {
    if (isFull(stack)) {
        printf("Stack overflow: Cannot push element %d, stack is full.\n", element);
    } else {
        stack->top++;
        stack->data[stack->top] = element;
        printf("Pushed element %d onto the stack.\n", element);
    }
}
int pop(Stack *stack) {
    if (isEmpty(stack)) {
        printf("Stack underflow: Cannot pop from empty stack.\n");
        return -1;
    } else {
```

Output

Clear

```
/tmp/1IzyvXNSOA.o
Pushed element 1 onto the stack.
Pushed element 2 onto the stack.
Pushed element 3 onto the stack.
Top element of the stack: 3
Popped element 3 from the stack.
Popped element 2 from the stack.
Top element of the stack after pops: 1
Popped element 1 from the stack.
Stack underflow: Cannot pop from empty stack.


=== Code Execution Successful ===
```