

2023학년도 2학기 출석과제물

교 과 목 명 : C++ 프로그래밍

학 번 : 202234-366307

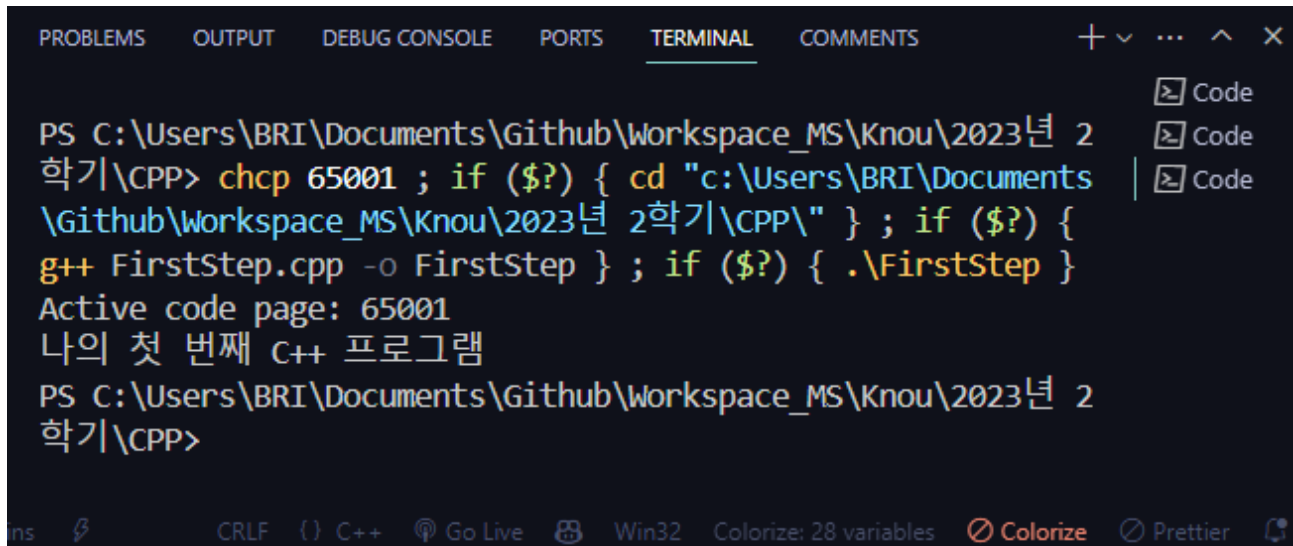
성 명 : 최문성

연 락 처 : 010-8010-6050

1. FirstStep.cpp

```
#include <iostream> // iostream 헤더 파일을 포함시키는 선행처리기 지시어
using namespace std; // using namespace 를 사용해서
// 명칭공간을 생략하고 std:: 없이 cout, endl 등을 사용 가능
int main() // 메인 함수
{
    cout << "나의 첫 번째 C++ 프로그램" << endl; // cout 객체를 이용하여
    화면에 텍스트를 콘솔에 출력
    return 0; // 0을 반환하고 종료
}
```

실행 결과



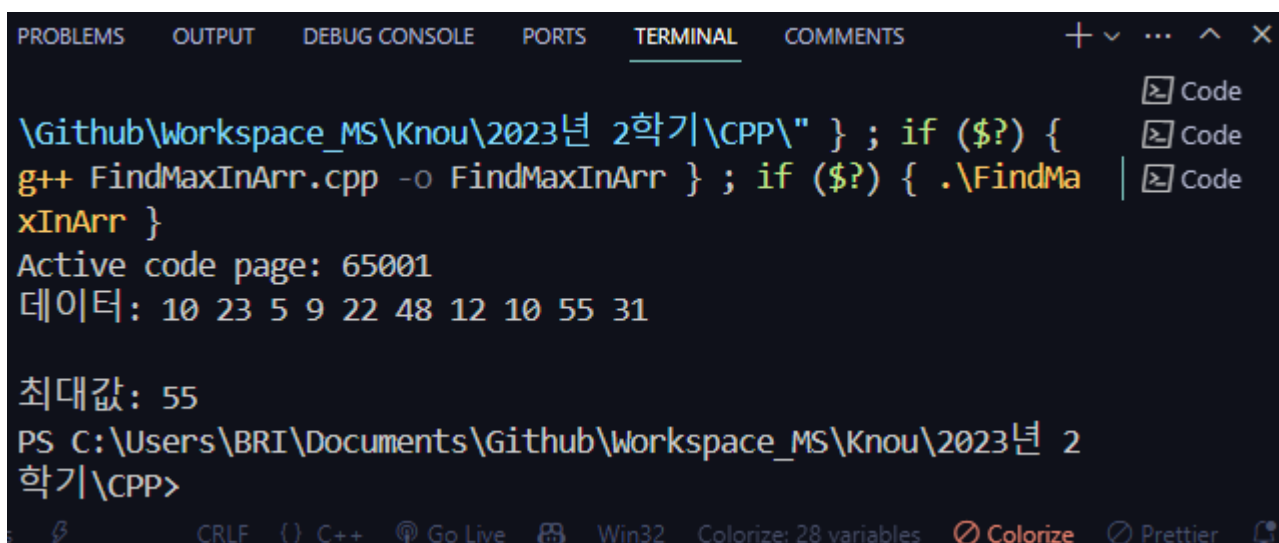
```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL COMMENTS + v ... ^ x
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP> chcp 65001 ; if ($?) { cd "c:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP\" } ; if ($?) { g++ FirstStep.cpp -o FirstStep } ; if ($?) { .\FirstStep }
Active code page: 65001
나의 첫 번째 C++ 프로그램
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP>
```

2. 10개의 값이 저장된 배열에서 최대값 구하기

```
#include <iostream> // iostream 헤더 파일을 포함시키는 선행처리 지시어
using namespace std; // using namespace 를 사용해서 명칭공간을 생략
int main()           // 메인 함수
{
    int data[10] = {10, 23, 5, 9, 22, 48, 12, 10, 55, 31}; // 10개의
    정수를 저장할 수 있는 data 배열을 선언하고 초기화
    int max = data[0]; // 최대값을
    저장할 변수 max 를 선언하고 data 배열의 첫 번째 값을 저장

    cout << "데이터: " << data[0]; // data 배열의 첫 번째 값을 출력
    for (int i = 1; i < 10; i++) // data 배열의 두 번째 값부터 마지막
    값까지 아래 블록을 9번 반복
    {
        cout << " " << data[i]; // data 배열의 i 번째 값에 공백을 더해서 출력
        if (data[i] > max) // 만약 data 배열의 i 번째 값이 max 보다 크면
            max = data[i]; // 최대값을 저장하는 변수 max 에 data 배열의
        i 번째 값을 저장
    }
    cout << endl
         << endl; // 줄바꿈
    cout << "최대값: " << max << endl; // 최대값을 출력
    return 0; // 0을 반환하고 종료
}
```

실행 결과



```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL COMMENTS + v ... ^ x
\Github\Workspace_MS\Knou\2023년 2학기\CPP\" } ; if ($?) {
g++ FindMaxInArr.cpp -o FindMaxInArr } ; if ($?) { .\FindMa |
xInArr }
Active code page: 65001
데이터: 10 23 5 9 22 48 12 10 55 31

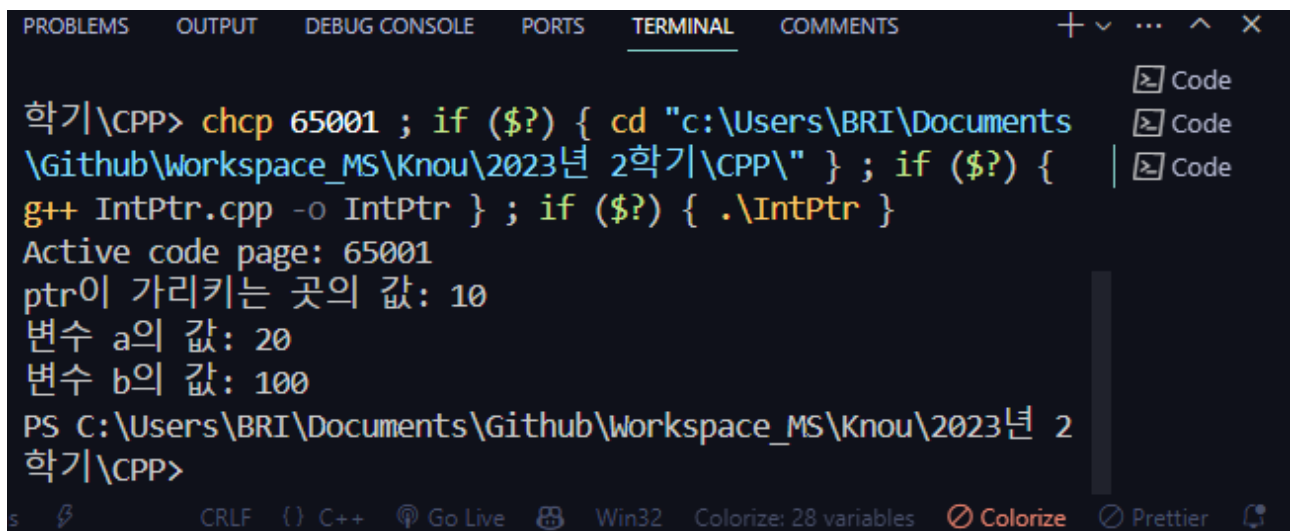
최대값: 55
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2
학기\CPP>
```

3. 포인터 예제

```
#include <iostream> // iostream 헤더 파일을 포함시키는 선행처리기 지시어
using namespace std; // using namespace 를 사용해서 명칭공간을 생략
int main()           // 메인 함수
{
    int a = 10, b = 100; // int 형 변수 a 와 b 를 선언, 각각 10과 100으로
    초기화
    int *ptr = &a;       // int 형 포인터 변수 ptr 을 선언, a 의 주소를 저장

    cout << "ptr 이 가리키는 곳의 값: " << *ptr << endl; // ptr 이 가리키는
    곳의 값을 출력. ptr 이 가리키는 곳은 a 이므로 10 출력
    *ptr = 20;                                     // ptr 이
    가리키는 곳에 20을 저장
    cout << "변수 a 의 값: " << a << endl;           // 변수 a 의 값을
    출력, a 는 20으로 변했으므로 20 출력
    ptr = &b;                                       // ptr 이
    가리키는 곳에 b 의 주소를 저장
    cout << "변수 b 의 값: " << *ptr << endl;       // ptr 이 가리키는
    곳의 값을 출력, b 는 100이므로 100 출력
    return 0;                                     // 0을 반환하고
    종료
}
```

실행 결과



```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL COMMENTS + v ... ^ x
학기\CPP> chcp 65001 ; if ($?) { cd "c:\Users\BRI\Documents
\Github\Workspace_MS\Knou\2023년 2학기\CPP\" } ; if ($?) {
g++ IntPtr.cpp -o IntPtr } ; if ($?) { .\IntPtr }
Active code page: 65001
ptr이 가리키는 곳의 값: 10
변수 a의 값: 20
변수 b의 값: 100
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2
학기\CPP>
```

4. 동적 메모리 할당 예제

```
#include <iostream> // iostream 헤더 파일을 포함시키는 선행처리기 지시어
using namespace std; // using namespace 를 사용해서 명칭공간을 생략하고
std:: 없이 cout, endl 을 사용할 수 있음
int main()           // 메인 함수
{
    int *intPtr;      // int 형 포인터 변수
    intPtr 선언
    intPtr = new int[4]; // int 4개를 저장할 배열
    intPtr 을 동적으로 할당, 그 주소를 포인터 intPtr 에 저장
    *intPtr = 10;      // 포인터 intPtr 이 가리키는
    곳에 10을 저장
    *(intPtr + 1) = 20; // 포인터 intPtr 이 가리키는
    곳의 다음 공간에 20을 저장
    intPtr[2] = 30;    // 포인터 intPtr 이 가리키는
    곳의 2번째 다음 공간에 30을 저장
    intPtr[3] = 40;    // 포인터 intPtr 이 가리키는
    곳의 3번째 다음 공간에 40을 저장
    for (int *p = intPtr, i = 0; i < 4; i++) // 포인터 p 에 intPtr 을
    대입하고 아래 블록을 4번 반복
        cout << *p++ << " "; // 포인터 p 가 가리키는 곳의
    값을 출력한 후 포인터 p 를 다음으로 이동
    cout << endl;      // 줄바꿈
    delete[] intPtr;   // 배열 intPtr 을 동적으로
    해제하여 메모리 반환
    return 0;          // 0을 반환하고 종료
}
```

실행 결과

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL COMMENTS + v ... ^ x
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP> chcp 65001 ; if ($?) { cd "c:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP\" } ; if ($?) { g++ DAlloc.cpp -o DAlloc } ; if ($?) { .\DAlloc }
Active code page: 65001
10 20 30 40
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2학기\CPP>
```

5. 참조 활용 예제

```
#include <iostream> // iostream 헤더 파일을 포함시키는 선행처리 지시어
using namespace std; // using namespace 를 사용해서 명칭공간을 생략
int main()           // 메인 함수
{
    int a = 10, b = 20; // int 형 변수 a 와
    b 를 선언, 각각 10과 20으로 초기화
    int &aRef = a;      // a 를 참조하는
    int 형 참조 변수 aRef 선언
    cout << "a 의 값: " << a << endl; // 변수 a 의 값을 출력
    cout << "aRef 가 참조하는 값: " << aRef << endl; // 참조 변수 aRef 가
    참조하는 값을 출력
    aRef = 100;         // 참조 변수 aRef 가
    참조하는 곳에 100을 저장
    cout << "a 의 값: " << a << endl; // 변수 a 의 값을
    출력, a 는 100으로 변했으므로 100출력
    aRef = b;           // 참조 변수 aRef 가
    참조하는 곳에 b 의 값을 저장
    cout << "a 의 값: " << a << endl; // 변수 a 의 값을
    출력, a 는 b 의 값이었던 20으로 변했으므로 20 출력
    return 0;           // 0을 반환하고 종료
}
```

실행 결과



```
g++ RecTest.cpp -o RecTest } ; if ($?) { .\RecTest }
Active code page: 65001
a의 값: 10
aRef가 참조하는 값: 10
a의 값: 100
a의 값: 20
PS C:\Users\BRI\Documents\Github\Workspace_MS\Knou\2023년 2
학기\CPP>
```