

[www.dut.bme.hu/courses/bap2](http://www.dut.bme.hu/courses/bap2)  
 Login using edut01 CT: 1.5 sec  
*berdineyem leírás*

MAT1, MAT2 = lecture  
 $10^2, 7^2, 10^2, 10^2 \geq 10^2 \geq 10^2$   
 $10, 10, 10, x, x$   
 $MAT1 + MAT2 \geq 100$   
 FINAL RESULT =  $\frac{MAT1 + MAT2 + 50}{2}$

**PROCESS OF COMPILE**

```

    graph TD
      A["SOURCE CODE"] --> B["PREPROCESSOR"]
      B --> C["COMPILER"]
      C --> D["LINKER"]
      D --> E["EXECUTABLE"]
  
```

**SOLUTION**

```

    graph LR
      A["#include <iostream.h>"] --> B["#include <iomanip.h>"]
      B --> C["#include <math.h>"]
      C --> D["#include <cmath.h>"]
      D --> E["#include <limits.h>"]
      E --> F["#include <stroct.h>"]
      F --> G["#include <complex.h>"]
      G --> H["#include <assert.h>"]
      H --> I["#include <string.h>"]
      I --> J["#include <vector.h>"]
      J --> K["#include <list.h>"]
      K --> L["#include <queue.h>"]
      L --> M["#include <stack.h>"]
      M --> N["#include <map.h>"]
      N --> O["#include <set.h>"]
      O --> P["#include <algorithm.h>"]
      P --> Q["#include <functional.h>"]
      Q --> R["#include <utility.h>"]
      R --> S["#include <new.h>"]
      S --> T["#include <exception.h>"]
      T --> U["#include <stdexcept.h>"]
      U --> V["#include <limitsf.h>"]
      V --> W["#include <limitsl.h>"]
      W --> X["#include <limitsr.h>"]
      X --> Y["#include <limitso.h>"]
      Y --> Z["#include <limitsn.h>"]
      Z --> AA["#include <limitsw.h>"]
      AA --> BB["#include <limitsfwd.h>"]
      BB --> CC["#include <limitslwd.h>"]
      CC --> DD["#include <limitsrwd.h>"]
      DD --> EE["#include <limitsowd.h>"]
      EE --> FF["#include <limitsnw.h>"]
      FF --> GG["#include <limitsnwd.h>"]
      GG --> HH["#include <limitswwd.h>"]
      HH --> II["#include <limitsnwdwd.h>"]
      II --> JJ["#include <limitsnwdwdwd.h>"]
      JJ --> KK["#include <limitsnwdwdwdwd.h>"]
      KK --> LL["#include <limitsnwdwdwdwdwd.h>"]
      LL --> MM["#include <limitsnwdwdwdwdwdwd.h>"]
      MM --> NN["#include <limitsnwdwdwdwdwdwdwd.h>"]
      NN --> OO["#include <limitsnwdwdwdwdwdwdwdwd.h>"]
      OO --> PP["#include <limitsnwdwdwdwdwdwdwdwdwd.h>"]
      PP --> QQ["#include <limitsnwdwdwdwdwdwdwdwdwdwd.h>"]
      QQ --> RR["#include <limitsnwdwdwdwdwdwdwdwdwdwdwd.h>"]
      RR --> SS["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      SS --> TT["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      TT --> UU["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      UU --> VV["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      VV --> WW["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      WW --> XX["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      XX --> YY["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      YY --> ZZ["#include <limitsnwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwdwd.h>"]
      ZZ --> AA
  
```

**PASSING PARAMETERS, REFERENCES**

```

    void f(int i) {
        int main() {
            int &x;
            x = 10;
            f(x);
            cout << x;
        }
    }
  
```

**returning References**

```

    int main() {
        int k, l;
        k = f(l);
        l = f(k);
        cout << k;
    }
  
```

**Hints & called? Advice, reference**

**DESTRUCTOR**

**copy constructor**

**INITIALIZATION LIST**

**CONSTANT**

**STATIC CLASS MEMBERS**

**INPUT/OUTPUT in C++**

**OPERATOR OVERLOADING**

**Complex Operator Overloading**

