```c
#include <stdio.h>


float test(float p1, int p2)
{
 float pom;
 pom = p1 + p2;
 return pom;
}

int main(int x, char *cc[])
{
 int i,j=1,k=2;
 float a,b=1,c=2;

 a = test (x,c);
 i = test (x,k);

 return a+i;
}
```

```
     1: #include <stdio.h>
     2:
     3:
     4: float test(float p1, int p2)
     5: {
00DE13E0 55              push        ebp
00DE13E1 8B EC           mov         ebp,esp
00DE13E3 81 EC CC 00 00 00 sub         esp,0CCh
00DE13E9 53              push        ebx
00DE13EA 56              push        esi
00DE13EB 57              push        edi
00DE13EC 8D BD 34 FF FF FF lea         edi,[ebp-0CCh]
00DE13F2 B9 33 00 00 00  mov         ecx,33h
00DE13F7 B8 CC CC CC CC  mov         eax,0CCCCCCCCh
00DE13FC F3 AB           rep stos    dword ptr
es:[edi]
     6:  float pom;
     7:  pom = p1 + p2;
00DE13FE DB 45 0C        fild        dword ptr [p2]
00DE1401 D8 45 08        fadd        dword ptr [p1]
00DE1404 D9 5D F8        fstp        dword ptr [pom]
     8:  return pom;
00DE1407 D9 45 F8        fld         dword ptr [pom]




     9: }
00DE140A 5F              pop         edi
00DE140B 5E              pop         esi
00DE140C 5B              pop         ebx
00DE140D 8B E5           mov         esp,ebp
00DE140F 5D              pop         ebp
00DE1410 C3              ret
    ----------------------------------




    10:
    11: int main(int x, char *cc[])
    12: {
00DE1420 55              push        ebp
00DE1421 8B EC           mov         ebp,esp
00DE1423 81 EC 08 01 00 00 sub         esp,108h
00DE1429 53              push        ebx
00DE142A 56              push        esi
00DE142B 57              push        edi
00DE142C 8D BD F8 FE FF FF lea         edi,[ebp-108h]
00DE1432 B9 42 00 00 00  mov         ecx,42h
00DE1437 B8 CC CC CC CC  mov         eax,0CCCCCCCCh
00DE143C F3 AB           rep stos    dword ptr
es:[edi]
```

```
     1: #include <stdio.h>
     2:
     3:
     4: float test(float p1, int p2)
     5: {




     6:  float pom;
     7:  pom = p1 + p2;




     8:  return pom;
00B51000 DB 44 24 08     fild        dword ptr
[esp+8]
00B51004 D8 44 24 04     fadd        dword ptr
[esp+4]
00B51008 D9 5C 24 08     fstp        dword ptr
[esp+8]
00B5100C D9 44 24 08     fld         dword ptr
[esp+8]
     9: }
00B51010 C3              ret




---------------------------------




    10:
    11: int main(int x, char *cc[])
    12: {
```

```
     1: #include <stdio.h>
     2:
     3:
     4: float test(float p1, int p2)
     5: {




     6:  float pom;
     7:  pom = p1 + p2;




     8:  return pom;




     9: }




---------------------------------




    10:
    11: int main(int x, char *cc[])
    12: {
```

```
    13:  int i,j=1,k=2;
00DE143E C7 45 EC 01 00 00 00 mov         dword ptr
[j],1
00DE1445 C7 45 E0 02 00 00 00 mov         dword ptr
[k],2
    14:  float a,b=1,c=2;
00DE144C D9 E8                fld1
00DE144E D9 5D C8             fstp        dword ptr [b]
00DE1451 D9 05 3C 58 DE 00 fld         dword ptr
[  real@40000000 (0DE583Ch)]
00DE1457 D9 5D BC             fstp        dword ptr [c]
    15:
    16:  a = test (x,c);
00DE145A D9 45 BC             fld         dword ptr [c]
00DE145D E8 7F FC FF FF   call
@ILT+220(__ftol2_sse) (0DE10E1h)
00DE1462 50                   push        eax
00DE1463 DB 45 08             fild        dword ptr [x]
00DE1466 51                   push        ecx
00DE1467 D9 1C 24             fstp        dword ptr [esp]
00DE146A E8 18 FC FF FF   call        test (0DE1087h)
00DE146F 83 C4 08             add         esp,8
00DE1472 D9 5D D4             fstp        dword ptr [a]
    17:  i = test (x,k);
00DE1475 8B 45 E0             mov         eax,dword ptr
[k]
00DE1478 50                   push        eax
00DE1479 DB 45 08             fild        dword ptr [x]
00DE147C 51                   push        ecx
00DE147D D9 1C 24             fstp        dword ptr [esp]
00DE1480 E8 02 FC FF FF   call        test (0DE1087h)
00DE1485 83 C4 08             add         esp,8
00DE1488 E8 54 FC FF FF   call
@ILT+220(__ftol2_sse) (0DE10E1h)
00DE148D 89 45 F8             mov         dword ptr
[i],eax
    18:

    19:  return a+i;
00DE1490 DB 45 F8             fild        dword ptr [i]
00DE1493 D8 45 D4             fadd        dword ptr [a]
00DE1496 E8 46 FC FF FF   call
@ILT+220(__ftol2_sse) (0DE10E1h)




  20: }
00DE149B 5F                   pop         edi
00DE149C 5E                   pop         esi
00DE149D 5B                   pop         ebx
00DE149E 81 C4 08 01 00 00 add          esp,108h
00DE14A4 3B EC                cmp         ebp,esp
```

```
    13:  int i,j=1,k=2;

    14:  float a,b=1,c=2;

    15:
    16:  a = test (x,c);

    17:  i = test (x,k);
00B51020 DB 44 24 04          fild        dword ptr
[esp+4]
00B51024 6A 02                push        2
00B51026 51                   push        ecx
00B51027 D9 1C 24             fstp        dword ptr [esp]
00B5102A E8 D1 FF FF FF   call        test (0B51000h)
00B5102F D9 5C 24 0C          fstp        dword ptr
[esp+0Ch]
00B51033 83 C4 08             add         esp,8

    18:

    19:  return a+i;
00B51036 D9 44 24 04          fld         dword ptr
[esp+4]
00B5103A D9 C0                fld         st(0)
00B5103C E8 0F 08 00 00   call        _ftol2_sse
(0B51850h)
00B51041 89 44 24 04          mov         dword ptr
[esp+4],eax
00B51045 DA 44 24 04          fiadd       dword ptr
[esp+4]
00B51049 E9 02 08 00 00   jmp         _ftol2_sse
(0B51850h)
    20: }
```

```
    13:  int i,j=1,k=2;

    14:  float a,b=1,c=2;

    15:
    16:  a = test (x,c);

    17:  i = test (x,k);
00A31000 DB 44 24 04          fild        dword ptr
[esp+4]
00A31004 DC 05 F8 20 A3 00 fadd         qword ptr
[__real@4000000000000000 (0A320F8h)]
00A3100A D9 5C 24 04          fstp        dword ptr
[esp+4]

    18:

    19:  return a+i;
00A3100E D9 44 24 04          fld         dword ptr
[esp+4]
00A31012 D9 C0                fld         st(0)
00A31014 E8 07 08 00 00   call        _ftol2_sse
(0A31820h)
00A31019 89 44 24 04          mov         dword ptr
[esp+4],eax
00A3101D DA 44 24 04          fiadd       dword ptr
[esp+4]
00A31021 E9 FA 07 00 00   jmp         _ftol2_sse
(0A31820h)
    20: }
```

00DE14A6 E8 A9 FC FF FF call @ILT+335(__RTC_CheckEsp) (0DE1154h) 00DE14AB 8B E5 mov esp,ebp 00DE14AD 5D pop ebp 00DE14AE C3 ret		
Přeloženo jako debug – bez optimalizací	Přeloženo jako release - vypnuto automatické nahrazování funkcí jejich inline variantou (optimalizace – Inline function expansion : Only __inline (**/Ob1**))	Přeloženo jako release

Žlutá – efektivní / nutný kód

Zelená – režie spojená s voláním funkcí jako podprogramu

Modrá – režie spojená s runtime testy (= testy za běhu programu)