# Παραδείγματα Ενδιάμεσου Κώδικα

Διαλέξεις στο μάθημα: Μεταφραστές Γεώργιος Μανής

TMHMA MHXANIKΩN H/Y & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

```
program ex1()
{ var b,c,g;
   function P1(in X,inout Y)
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
  var b,c,g;
   function P1(in X,inout Y)
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
  var b,c,g;
   function P1(in X,inout Y)
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
{ var b,c,g;
   function P1(in X,inout Y)
   { Y:=Y-1;
     if (X=1)
      return(X);
     else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
1: begin_block P1 _ _
program ex1()
                                    2: - Y 1 T 0
{ var b,c,g;
   function P1(in X,inout Y)
                              3: := T_0 _ Y
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
1: begin_block P1 _ _
program ex1()
{ var b,c,g;
                                     2: - Y 1 T 0
   function P1(in X,inout Y)
                                  3: := T_0 _ Y
                                     4: = X 1 _{-}
   { Y:=Y-1;
    if (X=1)
                                     5: jump _ _ _
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
1: begin_block P1 _ _
program ex1()
{ var b,c,g;
                                     2: - Y 1 T 0
   function P1(in X,inout Y)
                                  3: := T_0 _ Y
                                     4: = X \ 1 \ 6
   { Y:=Y-1;
    if (X=1)
                                     5: jump _ _ _
      return(X);
                                     6: retv X
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
1: begin_block P1 _ _
program ex1()
                                     2: - Y 1 T_0
{ var b,c,g;
   function P1(in X,inout Y)
                                   3: := T_0 _ Y
                                     4: = X \ 1 \ 6
   { Y:=Y-1;
    if (X=1)
                                     5: jump _ _ _
      return(X);
                                     6: retv X
    else
                                     7: jump _ _ _
      return(P1(in X-1,inout Y));
   c := 10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
                                    1: begin_block P1 _ _
{ var b,c,g;
                                    2: - Y 1 T 0
   function P1(in X,inout Y)
                                 3: := T 0 Y
                                    4: = X \ 1 \ 6
   { Y:=Y-1;
    if (X=1)
                                    5: jump _ _ 8
      return(X);
                                    6: retv X
    else
                                    7: jump _ _ _
      return(P1(in X-1,inout Y)); 8: - X 1 T_1
                                    9: par T 1 CV
                                    10: par Y REF
   c := 10;
   b:=5;
                                    11: par T_2 RET _
   g:=P1(in c, inout b);
                                    12: call _ P1
                                    13: retv T_2 _ _
```

```
program ex1()
                                    1: begin_block P1 _ _
{ var b,c,g;
                                    2: - Y 1 T 0
   function P1(in X,inout Y)
                                 3: := T 0 Y
                                    4: = X \ 1 \ 6
   { Y:=Y-1;
    if (X=1)
                                    5: jump _ _ 8
      return(X);
                                    6: retv X
                                    7: jump _ _ 14
    else
      return(P1(in X-1, inout Y)); 8: - X 1 T_1
                                    9: par T_1 CV _
                                    10: par Y REF
   c := 10;
   b:=5;
                                    11: par T_2 RET _
   g:=P1(in c, inout b);
                                    12: call _ P1
                                    13: retv T_2 _ _
                                    14: end block P1
```

```
15: begin_block ex1 _ _
program ex1()
                                     16: := 10 c
{ var b,c,g;
   function P1(in X,inout Y)
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
                                    15: begin_block ex1 _ _
{ var b,c,g;
                                    16: := 10 c
   function P1(in X,inout Y)
                              17: := 5 _ b
   { Y:=Y-1;
    if (X=1)
      return(X);
    else
      return(P1(in X-1,inout Y));
   c:=10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
                                    15: begin_block ex1 _ _
{ var b,c,g;
                                    16: := 10 c
   function P1(in X,inout Y)
                                    17: := 5 b
   { Y:=Y-1;
                                    18: par c CV _
    if (X=1)
                                    19: par b REF _
      return(X);
                                    20: par T 3 RET
    else
                                    21: call _ P1
      return(P1(in X-1,inout Y)); 22: := T_3 _ g
   c := 10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program ex1()
                                   15: begin_block ex1 _ _
{ var b,c,g;
                                   16: := 10 c
   function P1(in X,inout Y)
                                   17: := 5 b
   { Y:=Y-1;
                                   18: par c CV _
    if (X=1)
                                   19: par b REF _
      return(X);
                                   20: par T 3 RET
    else
                                   21: call _ _ P1
      return(P1(in X-1,inout Y)); 22: := T_3 _ g
                                   23: halt
                                   24: end block ex1
   c := 10;
   b:=5;
   g:=P1(in c, inout b);
```

```
program exams
  declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 and b<5)
   { if (t=1) c:=2;
    else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
        while(b=1)
         c:=2;
```

```
program exams
  declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 and b<5)
   { if (t=1) c:=2;
    else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
        while(b=1)
         c:=2;
```

```
program exams
   declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 and b<5)
   { if (t=1) c:=2;
    else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
        while(b=1)
         c:=2;
```

```
1: begin_block exams _ _
program exams
  declareint c,a,b,t enddeclare 2: := 1 _ a
   a:=1;
   while (a+b<1 \text{ and } b<5)
   { if (t=1) c:=2;
     else if (t=2) c:=4;
          else c:=0;
    while (a<1)
      if (a=2)
        while(b=1)
         c:=2;
```

```
program exams
  declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 and b<5)</pre>
   { if (t=1) c:=2;
     else if (t=2) c:=4;
          else c:=0;
     while (a<1)
       if (a=2)
        while(b=1)
         c := 2;
```

```
1: begin_block exams _ _ _
2: := 1 _ a
3: + a b T_1
4: < T_1 1 _
5: jump _ _ _
```

```
program exams
  declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 and b<5)
   { if (t=1) c:=2;
    else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
       while(b=1)
         c:=2;
```

```
1: begin_block exams _ _ _
2: := 1 _ a
3: + a b T_1
4: < T_1 1 6
5: jump _ _ _ _
6: < b 5 _
7: jump _ _ _
```

```
1: begin_block exams _ _
program exams
                                 2: := 1 _ a
  declareint c,a,b,t enddeclare
                                     3: + a b T 1
   a:=1;
   while (a+b<1 \text{ and } b<5)
                                     4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                     6: < b 5 8
         else c:=0;
                                     7: jump _ _ _
    while (a<1)
                                     8: = t 1 _
      if (a=2)
                                     9: jump _ _ _
        while(b=1)
         c:=2;
```

```
1: begin_block exams _ _
program exams
  declareint c,a,b,t enddeclare
                                  2: := 1 a
                                     3: + a b T 1
   a:=1;
   while (a+b<1 \text{ and } b<5)
                                     4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                     6: < b 5 8
         else c:=0;
                                     7: jump _ _ _
                                     8: = t 1 10
    while (a<1)
      if (a=2)
                                     9: jump _ _ _
                                     10: := 2 _ c
        while(b=1)
         c := 2;
```

```
program exams
                                     1: begin_block exams _ _
  declareint c,a,b,t enddeclare
                                 2: := 1 a
                                     3: + a b T 1
   a:=1;
   while (a+b<1 \text{ and } b<5)
                                     4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                     6: < b 5 8
         else c:=0;
                                     7: jump _ _ _
    while (a<1)
                                     8: = t 1 10
      if (a=2)
                                     9: jump _ _ _
        while(b=1)
                                     10: := 2 _ c
                                     11: jump _ _ _
         c := 2;
```

```
1: begin_block exams _ _
program exams
 declareint c,a,b,t enddeclare
                                 2: := 1 a
   a:=1;
                                     3: + a b T 1
   while (a+b<1 \text{ and } b<5)
                                    4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                     6: < b 5 8
         else c:=0;
                                     7: jump _ _ _
    while (a<1)
                                     8: = t \ 1 \ 10
      if (a=2)
                                     9: jump _ _ 12
        while(b=1)
                                     10: := 2 c
                                     11: jump _ _ _
         c := 2;
                                     12: = t 2 _
                                     13: jump _ _ _
```

```
1: begin_block exams _ _
program exams
                                 2: := 1 _ a
 declareint c,a,b,t enddeclare
   a:=1;
                                     3: + a b T 1
   while (a+b<1 \text{ and } b<5)
                                    4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                     6: < b 5 8
         else c:=0;
                                     7: jump _ _ _
    while (a<1)
                                     8: = t \ 1 \ 10
      if (a=2)
                                     9: jump 12
       while(b=1)
                                     10: := 2 c
         c := 2;
                                     11: jump _ _ _
                                     12: = t 2 14
                                     13: jump _ _ _
                                     14: := 4 _ c
```

```
1: begin_block exams _ _
program exams
                                 2: := 1 _ a
 declareint c,a,b,t enddeclare
   a:=1;
                                     3: + a b T 1
   while (a+b<1 \text{ and } b<5)
                                   4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                    6: < b 5 8
         else c:=0;
                                    7: jump _ _ _
                                     8: = t 1 10
    while (a<1)
      if (a=2)
                                     9: jump 12
       while(b=1)
                                     10: := 2 c
        c := 2;
                                     11: jump _ _ _
                                     12: = t 2 14
                                     13: jump _ _ _
                                     14: := 4 c
                                     15 jump _ _ _
```

```
1: begin_block exams _ _
program exams
{ declareint c,a,b,t enddeclare
                                 2: := 1 a
                                     3: + a b T 1
   a:=1;
   while (a+b<1 \text{ and } b<5)
                                   4: < T_1 1 6
   { if (t=1) c:=2;
                                     5: jump _ _ _
    else if (t=2) c:=4;
                                    6: < b 5 8
         else c:=0;
                                    7: jump _ _ _
    while (a<1)
                                    8: = t \ 1 \ 10
      if (a=2)
                                     9: jump 12
       while(b=1)
                                     10: := 2 c
        c := 2;
                                     11: jump _ _ _
                                     12: = t 2 14
                                     13: jump 16
                                     14: := 4 c
                                     15 jump _ _ _
                                     16 := 0 _ c
```

```
program exams
{ declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 \text{ and } b<5)
   { if (t=1) c:=2;
     else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
       while(b=1)
         c := 2;
```

```
2: := 1 a
3: + a b T 1
4: < T_1 1 6
5: jump _ _ _
6: < b 5 8
                17: < a 1 _
7: jump _ _ _
                18: jump _ _ _
8: = t 1 10
9: jump _ _ 12
10: := 2 c
11: jump _ _ 17
12: = t 2 14
13: jump 16
14: := 4 c
15 jump 17
16 := 0 c
```

1: begin\_block exams \_ \_

```
program exams
{ declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 \text{ and } b<5)
   { if (t=1) c:=2;
     else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
       while(b=1)
         c := 2;
```

```
1: begin_block exams _ _
2: := 1 a
3: + a b T 1
4: < T_1 1 6
5: jump _ _ _
6: < b 5 8
                17: < a 1 19
7: jump _ _ _
                18: jump _ _ _
8: = t 1 10
                19: = a 2 _
9: jump _ _ 12
                20: jump
10: := 2 c
11: jump _ _ 17
12: = t 2 14
13: jump 16
14: := 4 c
15 jump 17
16 := 0 c
```

```
program exams
{ declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 \text{ and } b<5)
   { if (t=1) c:=2;
     else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
       while(b=1)
         c := 2;
```

```
1: begin_block exams _ _
2: := 1 a
3: + a b T 1
4: < T 1 1 6
5: jump _ _
6: < b 5 8
                17: < a 1 19
7: jump _ _ _
                18: jump _ _ _
8: = t 1 10
                19: = a 2 21
9: jump _ _ 12
               20: jump _ _ _
10: := 2 c
                21: = b 1
11: jump _ _ 17
               22: jump _ _ _
12: = t 2 14
13: jump 16
14: := 4 c
15 jump 17
16 := 0 c
```

```
program exams
{ declareint c,a,b,t enddeclare
   a:=1;
   while (a+b<1 \text{ and } b<5)
   { if (t=1) c:=2;
    else if (t=2) c:=4;
         else c:=0;
    while (a<1)
      if (a=2)
       while(b=1)
         c:=2;
```

```
1: begin_block exams _ _
2: := 1 a
3: + a b T 1
4: < T_1 1 6
5: jump _ _
6: < b 5 8
              17: < a 1 19
7: jump _ _ _
              18: jump _ _ _
8: = t 1 10
              19: = a 2 21
9: jump _ _ 12
             20: jump
10: := 2 c
             21: = b 1 23
12: = t 2 14
              23: := 2 c
13: jump 16
14: := 4 c
15 jump 17
16 := 0 c
```

```
program exams
                               7: jump _ _ 28
{ declareint c,a,b,t enddeclare
                               8: = t 1 10
  a:=1;
                               9: jump _ _ 12
  while (a+b<1 \text{ and } b<5)
                               10: := 2 c
  { if (t=1) c:=2;
                               11: jump _ _ 17
    else if (t=2) c:=4;
                               12: = t 2 14
        else c:=0;
                              13: jump _ _ 16
    while (a<1)
                               14: := 4 c
     if (a=2)
                               15 jump _ _ 17
      while(b=1)
                               16 := 0 _ c
       c := 2;
```

```
3: + a b T 1
4: < T 1 1 6
5: jump _ _ 28
6: < b 5 8
                17: < a 1 19
                18: jump _ _ 27
                19: = a 2 21
                20: jump 26
                21: = b \ 1 \ 23
                22: jump 25
                23: := 2 c
                24: jump 21
                25: jump _ _ 26
                26: jump _ 17
                27: jump 3
                28: halt
                29: end block exams
```

1: begin\_block exams \_ \_

2: := 1 a

# Παράδειγμα 2 Ισοδύναμο σε C

```
int main()
 int a,b,T 1,t,c;
 L 1:
 L 2: a=1; //(:=, 1, ..., a)
 L 3: T l=a+b; //(+, a, b, T 1)
 L 4: if (T 1<1) goto L 6; //(<, T 1, 1, 6)
 L 5: goto L 28; //(jump, , , 28)
 L 6: if (b<5) goto L 8; //(<, b, 5, 8)
 L 7: goto L 28; //(jump, , , 28)
 L 8: if (t==1) goto L 10; //(=, t, 1, 10)
 L 9: goto L 12; //(jump, , , 12)
 L 10: c=2; //(:=, 2, , c)
 L 11: goto L 17; //(jump, , , 17)
 L 12: if (t==2) goto L 13; //(=, t, 2, 14)
 L 13: goto L 16; //(jump, , , 16)
 L 14: c=4; //(:=, 4, , c)
 L 15: goto L 17; //(jump, , , 17)
 L 16: c=0; //(:=, 0, , c)
 L 17: goto L 19; //(jump, , , 19)
 L 18: goto L 27; //(jump, , , 27)
 L 19: if (a==2) goto L 21; //(=, a, 2, 21)
 L 20: goto L 25; //(jump, , , 26)
 L 21: if (b==1) goto L 23; //(=, b, 1, 23)
 L 22: goto L 25; //(jump, , , 25)
 L 23: c=2; // (:=, 2, , c)
 L 24: goto L 21; //(jump, , , 21)
 L 25: goto L 26; //(jump, ,
 L 26: goto L 17; //(jump, , , 17)
 L 27: goto L 3; //(jump, , , 3)
 L 28: {}
```

```
program max()
{    var a,b,c,d,e;
    function max(in x, in y)
    {       if (x>y)
            return(x);
        else
            return(y);
    }

    e:=max(in max(in a, in b),
            in max(in c, in d));
}
```

```
program max()
{    var a,b,c,d,e;
    function max(in x, in y)
    {       if (x>y)
            return(x);
        else
            return(y);
    }

    e:=max(in max(in a, in b),
            in max(in c, in d));
}
```

```
0: begin_block MAX _ _ _
1: > x y 3
2: jump _ _ 5
3: retv x _ _
4: jump _ _ 6
5: retv y _ _
6: end_block MAX _ _
```

```
program max()
{    var a,b,c,d,e;
    function max(in x, in y)
    {       if (x>y)
            return(x);
        else
            return(y);
    }

    e:=max(in max(in a, in b),
            in max(in c, in d));
}
```

```
8: begin_block max _ _ _
9: par a CV _
10: par b CV
11: par T_1 RET _
12: call _ _ max
```

```
program max()
                                    8: begin_block max _ _
  var a,b,c,d,e;
                                    9: par a CV
   function max(in x, in y)
                                    10: par b CV
                                    11: par T_1 RET _
    { if (x>y)
          return(x);
                                    12: call _ _ max
       else
                                    13: par c CV
          return(y);
                                    14: par d CV _
                                    15: par T_2 RET _
                                    16: call _ _ max
    e:=max(in max(in a, in b),
          in max(in c, in d));
```

```
program max()
                              8: begin_block max _ _
 var a,b,c,d,e;
                              9: par a CV
   function max(in x, in y)
                              10: par b CV
   { if (x>y)
                              11: par T_1 RET _
        return(x);
                              12: call max
      else
                              13: par c CV
        return(y);
                              14: par d CV
                              15: par T 2 RET
                              16: call max
   e:=max(in max(in a, in b),
                              17: par T 1 CV
         19: par T_3 RET _
                              20: call _ _ max
```

```
program max()
                                 8: begin_block max _ _
  var a,b,c,d,e;
                                 9: par a CV
   function max(in x, in y)
                                 10: par b CV
   { if (x>y)
                                 11: par T_1 RET _
         return(x);
                                 12: call max
       else
                                 13: par c CV
         return(y);
                                 14: par d CV
                                 15: par T 2 RET
                                 16: call max
   e:=max(in max(in a, in b),
                                 17: par T 1 CV
          in max(in c, in d));
                             18: par T_2 CV _
                                 19: par T_3 RET _
                                 20: call max
                                 21: := T 3 e
                                 22: halt
                                 23: end_block max _
```

Ευχαριστώ