



# Διάλεξη 5: Φροντιστήριο για Συμβολοσειρές (Strings), Δείκτες (pointers) και Πίνακες Δεικτών

---

**Διδάσκων: Παναγιώτης Ανδρέου**

# Ερώτηση

Which one of the following variable names is NOT valid?

1. **Choice 1:**

go\_cart

2. **Choice 2:**

go4it

3. **Choice 3:**

4season



4. **Choice 4:**

run4

5. **Choice 5:**

\_what

# Ερώτηση

Which one of the following is NOT a valid C identifier?

1. Choice 1:

\_\_\_S

2. Choice 2:

1\_\_\_



3. Choice 3:

\_\_\_1

4. Choice 4:

\_\_\_

5. Choice 5:

S\_\_\_

# Ερώτηση

Which one of the following is NOT a valid identifier?

1. **Choice 1:**

`__ident`

2. **Choice 2:**

`while`



3. **Choice 3:**

`bigNumber`

4. **Choice 4:**

`g42277`

5. **Choice 5:**

`peaceful_in_space`

# Ερώτηση

Which one of the following is true for identifiers that begin with an underscore?

**1. Choice 1:**

They are generally treated differently by preprocessors and compilers from other identifiers.

**2. Choice 2:**

They are case-insensitive.

**3. Choice 3:**

They are reserved for usage by standards committees, system implementers, and compiler engineers.

**4. Choice 4:**



Applications programmers are encouraged to employ them in their own code in order to mark certain symbols for internal usage.

**5. Choice 5:**

They are deprecated by Standard C and are permitted only for backward compatibility with older C libraries.

# Ερώτηση

```
int x =  
2 * 3 +  
4 * 5;
```

What value will x contain in the sample code above?

1. **Choice 1:**  
22
2. **Choice 2:**  
26
3. **Choice 3:**  
46
4. **Choice 4:**  
50
5. **Choice 5:**  
70



# Ερώτηση

```
int i = 4;  
int x = 6;  
double z;  
z = x / i;  
printf("z=%.2f\n", z);
```

What will print when the sample code above is executed?

1. **Choice 1:**

z=0.00

2. **Choice 2:**

z=1.00



3. **Choice 3:**

z=1.50

4. **Choice 4:**

z=2.00

5. **Choice 5:**

z=NULL

# Ερώτηση

```
int z,  
    x=5,  
    y=-10,  
    a=4,  
    b=2;  
  
z = x++ - --y * b / a;
```

What number will z in the sample code above contain?

1. **Choice 1:**  
5
2. **Choice 2:**  
6
3. **Choice 3:**  
10
4. **Choice 4:**  
11
5. **Choice 5:**  
12





# Ερώτηση

When applied to a variable, what does the unary "&" operator yield?

**1. Choice 1:**

The variable's value

**2. Choice 2:**

The variable's binary form

**3. Choice 3:**

The variable's address



**4. Choice 4:**

The variable's format

**5. Choice 5:**

The variable's right value

# Ερώτηση

In a C expression, how is a logical AND represented?

1. **Choice 1:**

@@

2. **Choice 2:**

||

3. **Choice 3:**

.AND.

4. **Choice 4:**

&&

5. **Choice 5:**

.AND



# Ερώτηση

Which one of the following will read a character from the keyboard and will store it in the variable c?

1. **Choice 1:**

```
c = getc();
```

2. **Choice 2:**

```
getc( &c );
```

3. **Choice 3:**

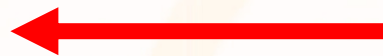
```
c = getchar( stdin );
```

4. **Choice 4:**

```
getchar( &c )
```

5. **Choice 5:**

```
c = getchar();
```



# Ερώτηση

**"My salary was increased by 15%!"**

Select the statement which will EXACTLY reproduce the line of text above.

**1. Choice 1:**

```
printf("\My salary was increased by 15/%!\\"\n");
```

**2. Choice 2:**

```
printf("My salary was increased by 15%!\n");
```

**3. Choice 3:**

```
printf("My salary was increased by 15%'!\n");
```

**4. Choice 4:**

```
printf("\My salary was increased by 15%%!\\"\n");
```

**5. Choice 5:**

```
printf("\My salary was increased by 15%'!\\"\n");
```



# Ερώτηση

Which one of the following printf() format specifiers indicates to print a double value in decimal notation, left aligned in a 30-character field, to four (4) digits of precision?

1. **Choice 1:**

`%-30.4e`

2. **Choice 2:**

`%4.30e`

3. **Choice 3:**

`%-4.30f`

4. **Choice 4:**

`%-30.4f`

5. **Choice 5:**

`%#30.4f`



# Ερώτηση

```
c = getchar();
```

What is the proper declaration for the variable c in the code above?

1. **Choice 1:**

```
char *c;
```

2. **Choice 2:**

```
unsigned int c;
```

3. **Choice 3:**

```
int c;
```

4. **Choice 4:**

```
unsigned char c;
```

5. **Choice 5:**

```
char c;
```



# Ερώτηση

```
int testarray[3][2][2]
=
{1,
2,
3,
4,
5,
6,
7,
8,
9,
10,
11,
12};
```

What value does testarray[2][1][0] in the sample code above contain?

- Choice 1:**  
3
- Choice 2:**  
5
- Choice 3:**  
7
- Choice 4:**  
9
- Choice 5:**  
11



# Ερώτηση

```
int i,j;
int ctr = 0;
int myArray[2][3];
for (i=0; i<3; i++)
  for (j=0; j<2; j++) {
    myArray[j][i] = ctr;
    ++ctr;
  }
```

What is the value of myArray[1][2]; in the sample code above?

1. **Choice 1:**

1

2. **Choice 2:**

2

3. **Choice 3:**

3

4. **Choice 4:**

4

5. **Choice 5:**

5





# Ερώτηση

```
char* myFunc (char
*ptr) {
    ptr += 3;
    return (ptr);
}

int main()
{
    char *x, *y;
    x = "HELLO";
    y = myFunc (x);
    printf ("y = %s \n",
y);
    return 0;
}
```

What will print when the sample code above is executed?


1. **Choice 1:**  
y = HELLO
2. **Choice 2:**  
y = ELLO
3. **Choice 3:**  
y = LLO
4. **Choice 4:**  
y = LO
5. **Choice 5:**  
x = O



# Ερώτηση

```
char *ptr;  
char myString[] =  
"abcdefg";  
ptr = myString;  
ptr += 5;
```

What string does ptr point to in the sample code above?

1. **Choice 1:**  
fg 
2. **Choice 2:**  
efg
3. **Choice 3:**  
defg
4. **Choice 4:**  
cdefg
5. **Choice 5:**  
None of the above

# Ερώτηση

```
int x = 3;  
if( x == 2 );  
    x = 0;  
if( x == 3 )  
    x++;  
else x += 2;
```

What value will x contain when the sample code above is executed?

1. **Choice 1:**

1

2. **Choice 2:**

2



3. **Choice 3:**

3

4. **Choice 4:**

4

5. **Choice 5:**

5

# Ερώτηση

```
int x = 0;  
for (x=1; x<4; x++);  
printf("x=%d\n", x);
```

What will be printed when the sample code above is executed?

1. **Choice 1:**

x=0

2. **Choice 2:**

x=1

3. **Choice 3:**

x=3

4. **Choice 4:**

x=4



5. **Choice 5:**

x=5

# Ερώτηση

```
x = 3, counter = 0;  
while ((x-1))  
{  
    ++counter;  
    x--;  
}
```

Referring to the sample code above, what value will the variable counter have when completed?

1. **Choice 1:**

0

2. **Choice 2:**

1

3. **Choice 3:**

2



4. **Choice 4:**

3


5. **Choice 5:**

4

# Ερώτηση

```
int x[] = {  
1, 4, 8, 5, 1, 4  
};  
  
int *ptr, y;  
  
ptr = x + 4;  
  
y = ptr - x;
```

What does y in the sample code above equal?

1. **Choice 1:**  
-3
2. **Choice 2:**  
0
3. **Choice 3:**  
4 
4. **Choice 4:**  
4 + sizeof( int )
5. **Choice 5:**  
4 \* sizeof( int )

# Ερώτηση

```
void myFunc (int x)
{
    if (x > 0)
        myFunc(--x);
    printf("%d, ",
x);
}

int main()
{
    myFunc(5);
    return 0;
}
```

What will the above sample code produce when executed?

1. **Choice 1:**  
0, 1, 2, 3, 4, 4,
2. **Choice 2:**  
4, 3, 2, 1, 0, 0,
3. **Choice 3:**  
5, 4, 3, 2, 1, 0,
4. **Choice 4:**  
0, 0, 1, 2, 3, 4, ←
5. **Choice 5:**  
0, 1, 2, 3, 4, 5,

# Ερώτηση

```
int x = 5;
int y = 2;
char op = '*';
switch (op)
{
    default : x += 1;
    case '+' : x += y;
    case '-' : x -= y;
}
```

After the sample code above has been executed, what value will the variable x contain?

1. **Choice 1:**

4

2. **Choice 2:**

5

3. **Choice 3:**

6

4. **Choice 4:**

7

5. **Choice 5:**

8





# Ερώτηση

```
#include <stdio.h>
void func()
{
    int x = 0;
    static int y = 0;
    x++; y++;
    printf( "%d -- %d\n", x, y
);
}

int main()
{
    func();
    func();
    return 0;
}
```

What will the code above print when it is executed?

1. **Choice 1:**

1 -- 1

1 -- 1

2. **Choice 2:**

1 -- 1

2 -- 1

3. **Choice 3:**

1 -- 1

2 -- 2

4. **Choice 4:**

1 -- 0

1 -- 0

5. **Choice 5:**

1 -- 1

1 -- 2




# Ερώτηση

```
short testarray[4][3]
= {
    {1},
    {2, 3},
    {4, 5, 6}
};

printf( "%d\n",
sizeof( testarray ) );
```

Assuming a short is two bytes long, what will be printed by the above code?

- Choice 1:**  
It will not compile because not enough initializers are given.
  - Choice 2:**  
6
  - Choice 3:**  
7
  - Choice 4:**  
12
  - Choice 5:**  
24
- 

# Ερώτηση

```
int x = 0;
for ( ; ; )
{
    if (x++ == 4)
        break;
    continue;
}
printf("x=%d\n", x);
```

What will be printed when the sample code above is executed?

1. **Choice 1:**

x=0

2. **Choice 2:**

x=1

3. **Choice 3:**

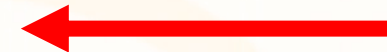
x=4

4. **Choice 4:**

x=5

5. **Choice 5:**

x=6



# Ερώτηση

```
#include <stdio.h>
int i;
void increment( int i )
{
    i++;
}

int main()
{
    for( i = 0; i < 10;
increment( i ) )
    {
    }
    printf("i=%d\n", i);
    return 0;
}
```

What will happen when the program above is compiled and executed?

- Choice 1:**  
It will not compile.
- Choice 2:**  
It will print out: i=9.
- Choice 3:**  
It will print out: i=10.
- Choice 4:**  
It will print out: i=11.
- Choice 5:** ←  
It will loop indefinitely.

# Ερώτηση

```
int i = 4;
switch (i)
{
    default:
        ;
    case 3:
        i += 5;
        if ( i == 8)
        {
            i++;
            if (i == 9) break;
            i *= 2;
        }
        i -= 4;
        break;
    case 8:
        i += 5;
        break;
}
printf("i = %d\n", i);
```

What will the output of the sample code above be?

1. **Choice 1:**

i = 5



2. **Choice 2:**

i = 8

3. **Choice 3:**

i = 9

4. **Choice 4:**

i = 10

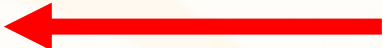
5. **Choice 5:**

i = 18

# Ερώτηση

```
int y[4] = {6, 7, 8, 9};  
int *ptr = y + 2;  
printf("%d\n",  
ptr[ 1 ] );
```

What is printed when the sample code above is executed?

1. **Choice 1:**  
6
  2. **Choice 2:**  
7
  3. **Choice 3:**  
8
  4. **Choice 4:**  
9
  5. **Choice 5:**  
The code will not compile.
- 

# Ερώτηση

```
char txt [20] = "Hello world!\0";
```

How many bytes are allocated by the definition above?


- Choice 1:**  
11 bytes
- Choice 2:**  
12 bytes
- Choice 3:**  
13 bytes
- Choice 4:**  
20 bytes
- Choice 5:**  
21 bytes



# Ερώτηση

```
long factorial (long x)
{
    ????
    return x * factorial(x - 1);
}
```

With what do you replace the ???? to make the function shown above return the correct answer?

- Choice 1:**  
if (x == 0) return 0;
- Choice 2:**  
return 1;
- Choice 3:**  
if (x >= 2) return 2;
- Choice 4:**  
if (x == 0) return 1; 
- Choice 5:**  
if (x == 1) return 1;



# Ερώτηση

```
char * dwarves [] = {  
    "Sleepy",  
    "Dopey" "Doc",  
    "Happy",  
    "Grumpy" "Sneezy",  
    "Bashful",  
};
```

How many elements does the array `dwarves` (declared above) contain? Assume the C compiler employed strictly complies with the requirements of Standard C.

1. **Choice 1:**

4

2. **Choice 2:**

5



3. **Choice 3:**

6

4. **Choice 4:**

7

5. **Choice 5:**

8

# Ερώτηση

```
char *buffer = "0123456789";  
char *ptr = buffer;  
ptr += 5;  
printf( "%s\n", ptr );  
printf( "%s\n", buffer );
```

What will be printed when the sample code above is executed?


1. **Choice 1:**  
0123456789  
56789
2. **Choice 2:**  
5123456789  
5123456789
3. **Choice 3:**  
56789  
56789
4. **Choice 4:**  
0123456789  
0123456789
5. **Choice 5:**  
56789  
0123456789



# Ερώτηση

```
int fibonacci (int n)
{
  switch (n)
  {
    default:
      return (fibonacci(n - 1)
+ fibonacci(n - 2));
    case 1:
    case 2:
  }
  return 1;
}
```

The function above has a flaw that may result in a **serious error** during some invocations. Which one of the following describes the deficiency illustrated above?

- Choice 1:**  For some values of n, the environment will almost certainly exhaust its stack space before the calculation completes.
- Choice 2:** An error in the algorithm causes unbounded recursion for all values of n.
- Choice 3:** A break statement should be inserted after each case. Fall-through is not desirable here.
- Choice 4:** The fibonacci() function includes calls to itself. This is not directly supported by Standard C due to its unreliability.
- Choice 5:** Since the default case is given first, it will be executed before any case matching n.

# Ερώτηση

```
int a [8] = { 0, 1, 2, 3 };
```

The definition of a above explicitly initializes its first four elements. Which one of the following describes how the compiler treats the remaining four elements?

1. **Choice 1:**

Standard C defines this particular behavior as implementation-dependent. The compiler writer has the freedom to decide how the remaining elements will be handled.

2. **Choice 2:**

The remaining elements are initialized to zero(0).



3. **Choice 3:**

It is illegal to initialize only a portion of the array. Either the entire array must be initialized, or no part of it may be initialized.

4. **Choice 4:**

As with an enum, the compiler assigns values to the remaining elements by counting up from the last explicitly initialized element. The final four elements will acquire the values 4, 5, 6, and 7, respectively.

5. **Choice 5:**

They are left in an uninitialized state; their values cannot be relied upon.

# Ερώτηση

```
int x[] = {1, 2, 3, 4, 5};
int u;
int *ptr = x;
????
for( u = 0; u < 5; u++ )
{
    printf("%d-", x[u]);
}
```

```
printf( "\n" );
```

Which one of the following statements could replace the ???? in the code above to cause the string 1-2-3-10-5- to be printed when the code is executed?

1. **Choice 1:**

```
*ptr + 3 = 10;
```

2. **Choice 2:**

```
*ptr[ 3 ] = 10;
```

3. **Choice 3:**

```
*(ptr + 3) = 10;
```

4. **Choice 4:**

```
(*ptr)[ 3 ] = 10;
```

5. **Choice 5:**

```
*(ptr[ 3 ]) = 10;
```



# Ερώτηση

```
int a=10,b;  
b=a++ + ++a;  
  
printf("%d,%d,%d,%d",  
b,  
a++,  
a,  
++a);
```

What will be the output when following code is executed?

1. **Choice 1:**  
12,10,11,13
2. **Choice 2:**  
22,10,11,13
3. **Choice 3:**  
12,12,12,12
4. **Choice 4:**  
22,12,13,14
5. **Choice 5:**  
22,13,13,14 ←