# EXAMEN FINAL CISCO ESTRUCTURA DE DATOS (73%)

#### **Question 1**

The following symbol may be (in certain circumstances) taken as an equivalent of:

```
a bracket

a tilde

a hash

a parenthesis
```

#### **Question 2**

Which of the given examples is contemporaty equivalent of the obsolete function declaration?

```
void f(x) float x; { return x * x; }

float f(float x) { return x * x; }

void f() float x; { return x * x; }

void f(float) x; { return x * x; }

void f(x) float; { return x * x; }
```

The ellipsis ( ) used in a function header, may be:

preceded by parameter of pointer type

surrounded by brackets

the first element on parameters list

0

the only element on parameters list

### **Question 4**

The so-called "calling convention" determines:

the way in which the arguments are put on and cleared off the stack

the way in which the programmer builds functions' interface

the possible function names

the possible argument names

A macro named va\_arg, comes from:



# Question 6

In the argument passing convention named "stdcall":

the invoker cleans the stack after return

the invokee cleans the stack before return

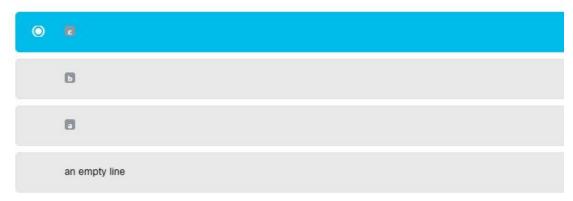
the stack is cleaned periodically by OS

the stack cleans its state itself

What is the expected output of the following code?

```
#include <stdarg.h>
#include <stdio.h>
int f(int n, ...) {
    va_list list;
    va_start(list,n);
    char c;
    while(va_arg(list,int) != 0)
    c = va_arg(list,int);
    va_end(list);
    return c;
}

int main(void) {
    printf("[%c]\n", f('a','b','c','\0'));
    return 0;
}
```

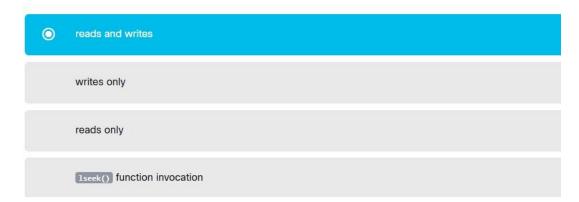


Each invocation of va\_start() must be matched by:

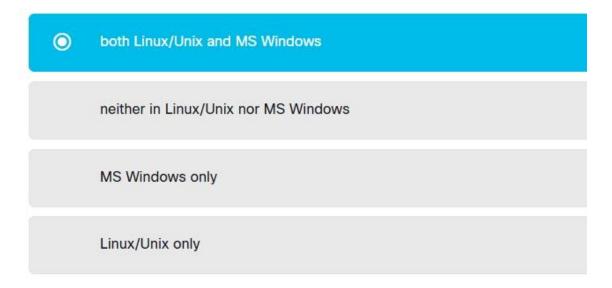


### Question 9

The "endline translation" occurs when the file is opened with oters flag and takes place during:



Symbolic links are available in:



### **Question 11**

Which of the expressions can be used to discover that [read()] invocation reached end of file?

```
read(fd, t, sizeof(t)) == 0

read(fd, t, sizeof(t)) > 0

read(fd, t, sizeof(t)) == sizeof(t)

read(fd, t, sizeof(t)) < 0
```

The file descriptor of value equal to 1:



### **Question 13**

The <code>lseek()</code> is able to move the file pointer beyond file's end.



The "st\_dev" field of "struct stat" in MS Windows environment reflects:

```
the drive number

device location

device vendor's name

32 bit long device identifier
```

### **Question 15**

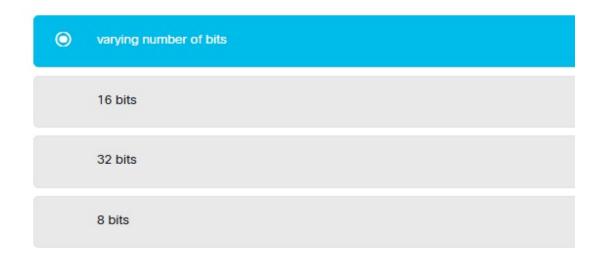
What should be placed instead of (2) to make the following snippet work properly?

```
if( <?> access("input", W_OK) ) {
   puts("File could not be written");
   exit(1);
}

NULL ==

NULL ==
```

To store UNICODE code points the UTF-8 uses:



#### **Question 17**

The westepy() function is:



What is the expected output of the following code?

```
#include <string.h>
#include >stdio.h>
int main(void) {
    char p[] = "10101", *q;
    int i = 0;
    q = strtok(p,"0");
    while(q) {
    i++;
    tq = strtok(MULL,"0");
    }
    printf("Xd\n",i);
    return 0;
}
```

#### Question 19

What is the expected output of the following code?

the code falls in an infinitive loop

```
#include <string.h>
#include >stdio.h>
int main(void) {
    int i, *p = &i;
    printf("%d\n", memcmp(&i,&p,4));
    return 0;
}

a number most probably different from zero

a number always less than zero

the output is unpredictable

Pantalla de Jacky
```

Dontalla da Jaclar

The result of the following expression:

```
depends on encoding system used by specific hardware platform

is always 
is always 
is always 
is always 
is always 8
```

#### Question 21

The memory block allocated by malloc() function:



#### **Question 22**

The value of the following expression:



What is possible output of the following program?

```
#include <stdio.h>
int main(int argc, char *argv[]) {
   puts(*argv);
   return 0;
}

a line containing program name

an empty line

a random string

zero

Pantalla de Jacky
```

#### **Question 24**

The string literal passed to Pragma keyword is the subject of:



#### **Question 25**

Which of the following main() function's headers is not valid?

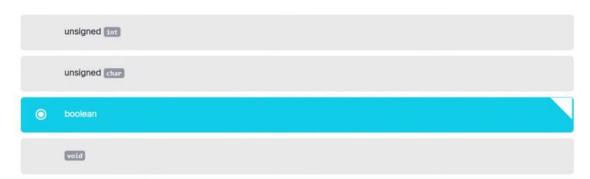
```
int main(int argc, char *argv[], char ***env[])

int main(int argc, char *argv[], char **env)

int main(int argc, char *argv[], char *env[])

int main(int argc, char *argv[])
```

The Bool keyword, introduced by C11, denotes a type being an equivalent of:



#### **Question 27**

Which of the following statements is true?

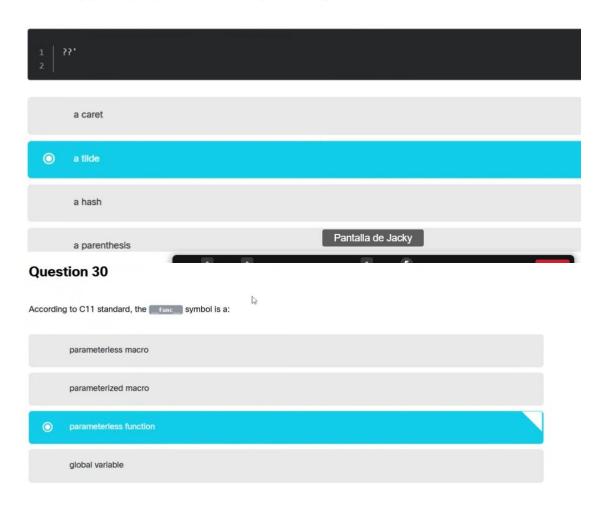


#### **Question 28**

A variadic function is a function which:

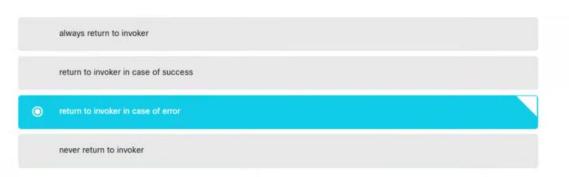


The following symbol may be (in certain circumstances) taken as an equivalent of:



#### Question 31

The [speen()] family functions:



A working process is identified by:



#### **Question 33**

The term "lock a mutex" being used in P-threads terminology:

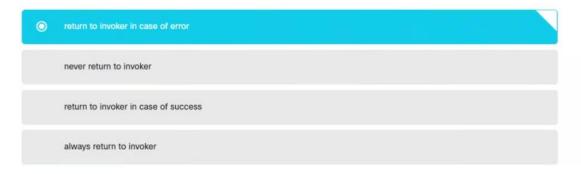


Assuming that the code was compiled and ran in Unix/Linux environment what is its expected output?

#### **Question 35**

The exec() family functions:

an empty line



The shortest part of a double value is:



#### **Question 37**

The length of a double typed variable:



#### Question 38

The FP SUBNORMAL flag marks a float value that:





#### Question 40

To input a 64 bit unsigned int value as hex number some would use the following portable specifier:



#### **Question 41**

B

The normalized float value:



The GMP library uses the pf t type to represent:



#### Question 43

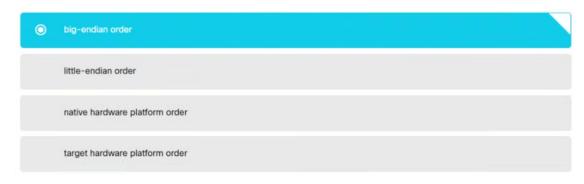
The UDP protocol:



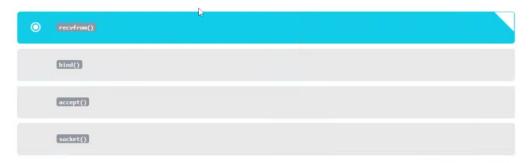
#### **Question 44**



The so-called "network order" is identical to:



The MSG\_PEEK flag is used in connection with:



#### **Question 46**

The regular IP address consists of:



#### **Question 47**

Using WinSock, comparing to the BSD sockets:

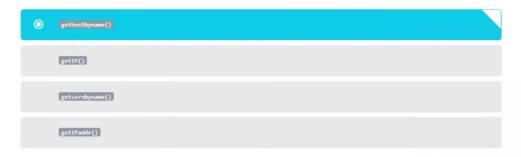


Which of the following is a proper bind() function header?



#### **Question 49**

Resolving a host's name based on given IP address can be done by invoking:



#### **Question 50**

The following declarator declares the fun symbol as:



The setsing() function performs its return sequence:

usually twice

more than twice

not more than once

#### Question 52

The following snippet:

int f(int i) {
 i++;
 if(i == 0) goto go\_to;
 return i;
}

int main(void) {
 f(0);
 go\_to:
 return 0;
}

will cause infinite loop

will cause runtime error

will cause compilation error

6

Select the true statement:

```
a const - volatile variable may be explicitly modified in the code containing its declaration

a const - volatile variable must not be implicitly modified by the background process

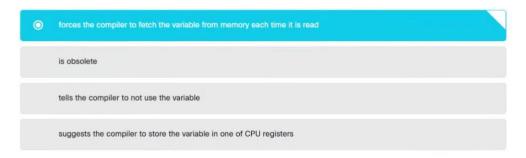
a compiler will try to store the const - volatile variable in CPU registers
```

#### **Question 54**

The following snippet:



The volatile specifier:



#### **Question 56**

A syntax used in the following snippet is named:

