

**CONCORSO PUBBLICO, PER ESAMI, A N.1 UNITÀ DI CATEGORIA D, POSIZIONE ECONOMICA D1, AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, PER LE ESIGENZE DEL DIPARTIMENTO DI MATEMATICA E APPLICAZIONI "RENATO CACCIOPPOLI" (DIPMATA) DELL'UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II (COD. RIF. 2233) INDETTO CON DECRETO DEL DIRETTORE GENERALE N. 1017 DEL 19/10/2022 DEL QUALE È STATO DATO AVVISO SULLA G.U. IV SERIE SPECIALE - CONCORSI ED ESAMI N.87 DEL 04/11/2022**

## QUESITI ESTRATTI ALLA PROVA ORALE DEL 18.01.2023

### PROVA ORALE n. 3

1. Il candidato illustri uso e caratteristiche di un Content Management System (CMS), fornendo anche qualche esempio.
2. Il candidato illustri e discuta di quali partizioni si ha bisogno per la installazione di un sistema GNU/Linux. Spieghi, inoltre, se è possibile che su di uno stesso computer siano installati ad esempio due diversi sistemi operativi, motivandone l'impossibilità, oppure motivandone la possibilità, indicando come.
3. Il candidato legga e traduca, spiegandone il senso, il seguente brano , tratto da Knuth, Donald: MMIX, :The Art of Computer Programming, Fascicle 1, pag. 63:  
First we break the whole program into a small number of pieces, which might be thought of temporarily as subroutines although they are called only once. These pieces are successively refined into smaller and smaller parts, having correspondingly simpler jobs to do. Whenever some computational task arises that seems likely to occur elsewhere or that has already occurred elsewhere, we define a subroutine (a real one) to do that job. We do not write the subroutine at this point; we continue writing the main program, assuming that the subroutine has performed its task. Finally, when the main program has been sketched, we tackle the subroutines in turn, trying to take the most complex subroutines first and then their sub-subroutines, etc. In this manner we will come up with a list of subroutines. The actual function of each subroutine has probably already changed several times, so that the first parts of our sketch will by now be incorrect; but that is no problem, since we are merely making a sketch. We now have a reasonably good idea about how each subroutine will be called and how general-purpose it should be. We should consider extending the generality of each subroutine, at least a little.

### PROVA ORALE n. 4

1. Linguaggi di programmazione compilati e linguaggi di programmazione interpretati. A partire da Fortran o C/C++ o Python, il candidato discuta di questo aspetto nella progettazione di programmi, con alcuni esempi.
2. Il candidato illustri gli strumenti che si possono utilizzare nella progettazione/realizzazione di un sito web dipartimentale, indicando a suo giudizio le principali funzionalità, esigenze, ecc...
3. Il candidato legga e traduca, spiegandone il senso, il seguente brano , tratto da Knuth, Donald: MMIX, :The Art of Computer Programming, Fascicle 1, pag. 64:  
Debugging is an art that needs much further study, and the way to approach it is highly dependent on the facilities available at each computer installation. A good start towards

effective debugging is often the preparation of appropriate test data. The most successful debugging techniques are typically designed and built into the program itself: Many of today's best programmers devote nearly half of their programs to facilitating the debugging process in the other half. The first half, which usually consists of fairly straightforward routines that display relevant information in a readable format, will eventually be of little importance, but the net result is a surprising gain in productivity. Another good debugging practice is to keep a record of every mistake made. Even though this will probably be quite embarrassing, such information is invaluable to anyone doing research on the debugging problem, and it will also help you learn how to cope with future errors.

Per ordine del Presidente

Il Segretario della Commissione

f.to dott.ssa Emma Di Marino