

Question 1: Introduction to Computer Programming

What is a compiler?
(2 points)

Cos'è un compilatore?
(2 punti)

A compiler is a software tool that translates a program written in a high-level programming language into an equivalent machine language program. It is composed of a lexical analyzer, a parser, a semantic analyzer, a code generator, and a code optimizer.

Un compilatore è uno strumento software che traduce un programma scritto in un linguaggio di programmazione di alto livello di astrazione in un programma equivalente in linguaggio macchina. Si compone di un analizzatore lessicale, un riconoscitore sintattico, un analizzatore semantico, un generatore di codice e un ottimizzatore di codice.

Question 2: Introduction to the Language ANSI C

What is the role of placeholders within format strings?
(2 points)

*Qual è il ruolo dei segnaposto all'interno delle stringhe formate?
(2 punti)*

Placeholders stand for data to be read or to be written. In the case of a scanf-like function, they represent data to be read and then to be assigned to the variables whose addresses are contained in the function invocation. In the case of a printf-like function, they represent data to be written on the basis of the evaluation of the expressions contained in the function invocation. The placeholders and the variables/expressions must coincide by number, order, and type.

I segnaposto stanno per dati da leggere o da scrivere. Nel caso di una funzione come scanf, essi rappresentano dati da leggere e poi da assegnare alle variabili i cui indirizzi sono contenuti nell'invocazione della funzione. Nel caso di una funzione come printf, essi rappresentano dati da scrivere sulla base della valutazione delle espressioni contenute nell'invocazione della funzione. I segnaposto e le variabili/espressioni debbono coincidere per numero, ordine e tipo.

Question 3: ANSI C Expressions

Explain the difference between symbolic constants and literal constants.
(2 points)

*Spiegare la differenza tra costanti simboliche e costanti in senso letterale.
(2 punti)*

A symbolic constant is defined via a `#define` directive, which provides an `int/double/char/string` value with a symbolic name that can be used wherever in the rest of the program. A literal constant is just an `int/double/char/string` value with no symbolic name associated with it. The use of symbolic constants increases program readability and maintainability.

Una costante simbolica è definita attraverso una direttiva `#define`, la quale dota un valore di tipo `int/double/char/stringa` di un nome simbolico che può essere usato ovunque nel resto del programma. Una costante in senso letterale è semplicemente un valore di tipo `int/double/char/stringa` senza nessun nome simbolico associato. L'uso di costanti simboliche incrementa la leggibilità e la mantenibilità dei programmi.

Question 4: ANSI C Statements

Describe syntax and semantics of the multiple assignment statement.
(2 points)

*Descrivere sintassi e semantica dell'istruzione di assegnamento multiplo.
(2 punti)*

The syntax of this statement is “var_1 = var_2 = ... = var_n = expr;”. The meaning is that first the expression expr is evaluated, then its value is stored into the memory locations associated with the variables var_n, ..., var_2, var_1 considered in this order.

La sintassi di questa istruzione è “var_1 = var_2 = ... = var_n = espr;”. Il significato è che prima viene valutata l'espressione espr, poi il suo valore viene collocato nelle locazioni di memoria associate alle variabili var_n, ..., var_2, var_1 considerate in quest'ordine.

Question 5: ANSI C Functions

Explain the difference between `argc` and `argv`.
(2 points)

*Spiegare la differenza tra `argc` e `argv`.
(2 punti)*

They represent the parameters of function main. While argc stores the number of strings (executable file name together with possible options and input or output files) occurring in the command given to start the execution of the program, argv stores the strings themselves.

Essi rappresentano i parametri della funzione main. Mentre argc contiene il numero di stringhe (nome del file eseguibile assieme a eventuali opzioni e file di input o output) presenti nel comando dato per avviare l'esecuzione del programma, argv contiene le stringhe stesse.

Question 6: ANSI C Data Types

What is a type constructor?
(2 points)

Cos'è un costruttore di tipo?
(2 punti)

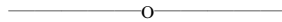
A type constructor is a mechanism through which a new type can be constructed on the basis of already existing types. The type constructors provided by the language ANSI C are enum, [] (for array types), struct, union, and * (for pointer types).

*Un costruttore di tipo è un meccanismo tramite il quale un nuovo tipo può essere costruito sulla base di tipi già esistenti. I costruttori di tipo messi a disposizione dal linguaggio ANSI C sono enum, [] (per i tipi array), struct, union e * (per i tipi puntatore).*

Question 7: Structured Programming and Correctness

What is a flow chart?
(2 points)

Cos'è uno schema di flusso?
(2 punti)

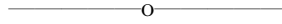


How is the meaning of a sequential program formalized?
(2 points)

Come viene formalizzato il significato di un programma sequenziale?
(2 punti)

A flow chart is a graphical representation of a program.

Uno schema di flusso è una rappresentazione grafica di un programma.



The meaning of a sequential program is formalized as a mathematical function that describes the input/output effect of the execution of the program by ignoring the intermediate computation states. By computation state we mean the memory contents at a certain point of the execution of the program.

Il significato di un programma sequenziale viene formalizzato mediante una funzione matematica che descrive l'effetto ingresso/uscita dell'esecuzione del programma ignorando gli stati intermedi della computazione. Per stato della computazione intendiamo il contenuto della memoria a un certo punto dell'esecuzione del programma.

Question 8: Linux/gvim/gcc/make/gdb

What is the Linux command man for?
(2 points)

A cosa serve il comando man di Linux?
(2 punti)

The Linux command `man` is a command line manual that provides information about the command specified as its argument in terms of syntax, semantics, and available options.

Il comando `man` di Linux è un manuale a linea di comando che fornisce informazioni sul comando specificato come suo argomento in termini di sintassi, semantica e opzioni disponibili.

Exercise 1: ANSI C Programming

Write a recursive ANSI C function that takes as input parameters a string and a character and returns the number of occurrences of the character within the string.

(7 points)

Scrivere una funzione ANSI C ricorsiva che ha come parametri di ingresso una stringa e un carattere e restituisce il numero di occorrenze del carattere all'interno della stringa.

(7 punti)

```
int count(char *s,
          char c)
{
    int n;

    if (s[0] == '\0')
        n = 0;
    else
    {
        n = (s[0] == c)?
            1:
            0;
        n += count(s + 1,
                  c);
    }
    return(n);
}
```

Exercise 2: ANSI C Verification

Prove via Hoare triples and Dijkstra rules that the following ANSI C function computes the maximum even number not greater than x :

(7 points)

```
int compute_max_even(int x)
{
    int z;

    if (x % 2 == 0)
        z = x;
    else
        z = x - 1;
    return(z);
}
```

Dimostrare mediante triple di Hoare e regole di Dijkstra che la precedente funzione ANSI C calcola il massimo numero pari non maggiore di x :

(7 punti)

Postcondition: $z = \max\{n \in \mathbb{Z} \mid n = n' \cdot 2 \wedge n \leq x\}$.

Observation: $x \% 2 == 0$ can be formalized as $x = x' \cdot 2$.

Observation: $x \% 2 != 0$ can be formalized as $x = x' \cdot 2 + 1$.

$(x = x' \cdot 2 \rightarrow (z = \max\{n \in \mathbb{Z} \mid n = n' \cdot 2 \wedge n \leq x\})_{z,x}) \equiv$
 $(x = x' \cdot 2 \rightarrow (x = \max\{n \in \mathbb{Z} \mid n = n' \cdot 2 \wedge n \leq x\})) \equiv true.$

$(x = x' \cdot 2 + 1 \rightarrow (z = \max\{n \in \mathbb{Z} \mid n = n' \cdot 2 \wedge n \leq x\})_{z,x-1}) \equiv$
 $(x = x' \cdot 2 + 1 \rightarrow (x - 1 = \max\{n \in \mathbb{Z} \mid n = n' \cdot 2 \wedge n \leq x\})) \equiv true.$

$(true \wedge true) \equiv true.$