

Question 1: introduction to computer programming

What is a compiler? (4 points).

Cos'è un compilatore? (4 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 static semantic checker (usually implementing a type system), an intermediate code generator, a code optimizer, and a code generator.

Un compilatore è uno strumento software che traduce un programma scritto in un linguaggio di programmazione ad alto livello di astrazione in un programma in linguaggio macchina equivalente. Si compone di un analizzatore lessicale, un riconoscitore sintattico, un controllore semantico statico (che di solito implementa un sistema di tipi), un generatore di codice intermedio, un ottimizzatore di codice e un generatore di codice.

Question 2: introduction to the language ANSI C

What is the role of the placeholders within the format strings? (4 points).

Qual è il ruolo dei segnaposto all'interno delle stringhe formate? (4 punti).

The 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 (respectively the expressions) must coincide by number, order, and type.

I segnaposto stanno per dati da leggere o da scrivere. Nel caso di una funzione tipo 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 tipo printf, essi rappresentano dati da scrivere sulla base della valutazione delle espressioni contenute nell'invocazione della funzione. I segnaposto e le variabili (rispettivamente le espressioni) debbono coincidere per numero, ordine e tipo.

Question 3: constants, variables, and expressions

Explain the difference between a symbolic constant and a literal constant (4 points).

Spiegare la differenza tra una costante simbolica e una costante in senso letterale (4 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/string` 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/string` senza nessun nome simbolico associato ad esso. L'uso di costanti simboliche incrementa la leggibilità e la mantenibilità dei programmi.

Question 4: statements

Describe syntax and semantics of the multi-assignment statement (4 points).

Descrivere sintassi e semantica dell'istruzione di assegnamento multiplo (4 punti).

The syntax of this statement is “ $\text{var1} = \text{var2} = \dots = \text{varn} = \text{expr};$ ”. The meaning is that first expression expr is evaluated, then its value is stored into the memory locations associated with the variables varn , ..., var2 , var1 considered in this order.

La sintassi di questa istruzione è “ $\text{var1} = \text{var2} = \dots = \text{varn} = \text{espr};$ ”. Il significato è che prima viene valutata l'espressione espr , poi il suo valore viene collocato nelle locazioni di memoria associate alle variabili varn , ..., var2 , var1 considerate in questo ordine.

Question 5: functions

Explain the difference between argc and argv (4 points).

Spiegare la differenza tra argc e argv (4 punti).

They represent the parameters of function main. While argc stores the number of strings (executable file name and possible options) 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 ed eventuali opzioni) presenti nel comando dato per avviare l'esecuzione del programma, argv contiene le stringhe stesse.

Question 6: data types

What is a type constructor? (4 points).

Cos'è un costruttore di tipo? (4 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 forniti dal linguaggio ANSI C sono enum, [] (per i tipi array), struct, union e * (per i tipi puntatori).*

Question 7: function development exercises

Write a recursive function that takes as input parameters a string and a character and returns as result the number of occurrences of the character within the string (6 points).

Scrivere una funzione ricorsiva che ha come parametri di ingresso una stringa e un carattere e restituisce come risultato il numero di occorrenze del carattere all'interno della stringa (6 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);
}
```