



CORSO DI PROGRAMMAZIONE: PROVA SCRITTA 6 FEBBRAIO 2001

COGNOME	NOME	MATRICOLA
---------	------	-----------

1. Qual è stato il principale contributo di John von Neumann nella storia dell'Informatica?

RISPOSTA:

2. Che cos'è un'espressione?

RISPOSTA:

3. Il comando seguente è corretto? Perché?

```
int x = 1;
x++ = x + 6;
```

RISPOSTA:

4. Che cosa produce in output la seguente porzione di codice?

```
int x, y;
for (x=0, y=5; x++ < y--; x=1){
    printf("(%d,%d)",x,y);
}
```

RISPOSTA:

5. Usando solo dichiarazioni, comandi di assegnamento e `while`, scrivere una sequenza di comandi equivalente al seguente comando `for`:

```
int i ,j;
int z = 1;
for (i=0, j=5; i+j < 10; i++)
    z *= j - i;
```

RISPOSTA:

-
6. Scrivere un'espressione booleana che testa se il valore di una variabile `x` di tipo `int` è un multiplo sia di 7 che di 13 che di 23.

RISPOSTA:

7. Per quali valori delle variabili intere `x, y` l'espressione booleana $((y < (x-1)) \ || \ (x < (y+1)))$ è vera?

RISPOSTA:

8. Scrivere la definizione di una procedura *ricorsiva* `scrivi(int n)` che dato numero intero `n` (con `n > 0`) scrive su una stessa riga i numeri pari strettamente positivi che sono minori o uguali a `n`.

RISPOSTA:

9. Per quali valori dei parametri attuali la seguente procedura termina correttamente (non per overflow dello stack)?

```
int strana(int a, int b, int c){
    if (a+b < c)
        return strana(a-1,b-1,c);
    else if (-c > c)
        return c;
    else
        return strana(a,b,c+1);
}
```

RISPOSTA:

10. La seguente espressione di inizializzazione è corretta? Perché?

```
int* i, y;
i = &y;
```

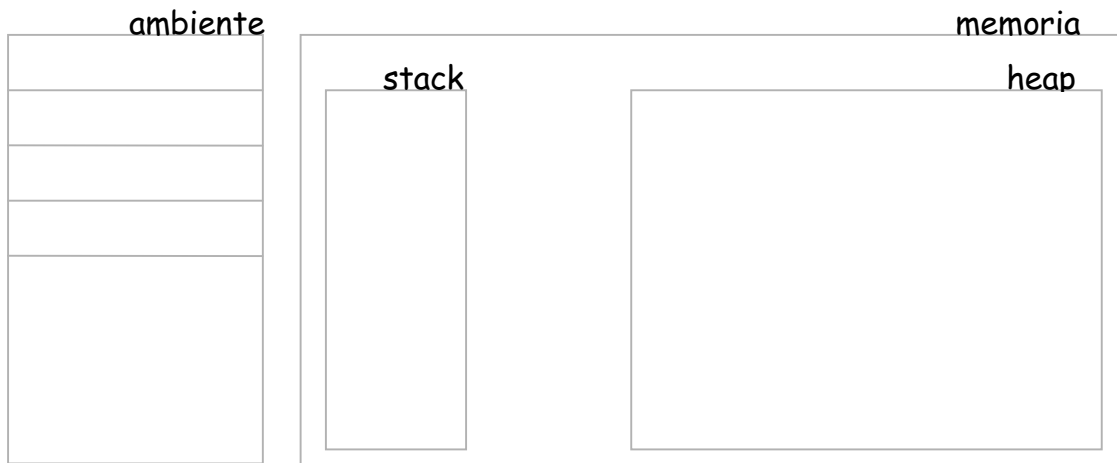
RISPOSTA:

-
11. Scrivere una serie di dichiarazioni e di comandi che allochino nello heap una lista semplice di 2 elementi. Gli elementi di questa lista devono essere dei record con tre campi: `alpha`, `beta` e `gamma`. Il campo `alpha` è di tipo `int`, e nei due elementi della lista questo campo deve essere inizializzato, rispettivamente, con i valori 4 e 8. Il tipo del campo `beta` è una tabella di `char`, e deve essere inizializzato in entrambi i casi con un array di dimensione 2 i cui elementi sono tutti uguali a 'w'. Il campo `gamma`, di tipo puntatore, deve puntare al prossimo elemento della lista (a NULL nel caso del secondo ed ultimo elemento).

RISPOSTA:

-
12. Utilizzando il modello ambiente-memoria, descrivere graficamente la situazione al termine dei comandi necessari a realizzare quanto richiesto dall'esercizio precedente.

RISPOSTA:



-
13. Scrivere la definizione della funzione `uguali`, che prende in ingresso una lista semplice di interi e restituisce 1 se tutti i valori contenuti nella lista sono uguali, e 0 altrimenti. Il tipo `Lista` e la dichiarazione della procedura sono i seguenti:

```
typedef struct Lista{ int num; struct Lista* prossimo;} Lista;
int uguali(Lista* lista);
```

RISPOSTA:

-
14. Ad una funzione che ha un solo parametro formale di tipo `int*` è possibile passare come parametro attuale
- a) il valore 0
 - b) il valore `NULL`
 - c) l'indirizzo di una qualsiasi variabile di tipo `int*`.

RISPOSTA:

- a)
 - b)
 - c)
-

15. Qual è il tipo della variabile `gianni` introdotta nell'ambiente dalla seguente dichiarazione?

```
double (*gianni[2])(double)
```

RISPOSTA:

16. Usando la variabile `gianni` dichiarata nell'esercizio precedente, assegnarle un valore.

RISPOSTA:

17. Cosa significa che una variabile può essere dichiarata `extern` o `static`?

RISPOSTA:

18. Scrivere la definizione di una funzione `void inverti(int tabella[], int n)` che prende in ingresso un array bidimensionale quadrato `tabella`, di dimensione `n*n`, e lo inverte rispetto alla diagonale principale.
Esempio:

1 2 3		1 4 7		1 2 3 4 5		1 6 11 16 21
4 5 6	diventa	2 5 8		6 7 8 9 10		2 7 12 17 22
7 8 9		3 6 9		11 12 13 14 15	diventa	3 8 13 18 23
				16 17 18 19 20		4 9 14 19 24
				21 22 23 24 25		5 10 15 20 25

RISPOSTA:
