

# INFORMATICA - CdL in FISICA

## Esercitazione dell'8 giugno 2011

### ESERCIZIO 1

Data la seguente definizione

```
void p(int x, int y)
{
    int a, b, *h, *k;
    a = (x + y);
    b = (x * y);
    h = &a;
    k = &b;
    if ((x % y) != 0)
    {   h = k;
        k = &a; }
    else
    {   *k = (*k) + (*h);
        *h = (*k) + (*h);   }
    printf("%d %d %d %d\n",a,b,*h,*k);
}
```

dire quali valori vengono stampati in corrispondenza delle due chiamate:

i) p(5, 5)

ii) p(8, 3)

### ESERCIZIO 2

Definire in modo **ricorsivo** una procedura che, dato un array di caratteri, la sua dimensione, un carattere  $c$  e un intero  $k$ , verifichi che il carattere  $c$  non compaia *piu' di k* volte nell'array.

Ad esempio, dato il vettore

'f'	'a'	'j'	'a'	'c'	'2'	'i'	'a'	'l'	'e'
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

con il carattere  $'a'$  e l'intero 3 la procedura deve restituire *true*, con il carattere  $'a'$  e l'intero 4 la procedura deve ancora restituire *true*, mentre con il carattere  $'a'$  e l'intero 2 la procedura deve restituire *false*.

### ESERCIZIO 3

Scrivere una funzione che verifica se la stringa passata come parametro e' palindroma oppure no.

### ESERCIZIO 4

Sia data la seguente definizione del tipo di dati *Lista*:

```
typedef struct nodo {int info; struct nodo *next;} Nodo;
typedef Nodo * Lista;
```

Scrivere la procedura *merge* con prototipo

```
void merge(Lista *result, Lista l1, Lista l2);
```

dove  $l1$  e  $l2$  sono due liste ordinate in modo crescente. Dopo l'invocazione di *merge* la variabile **result** deve puntare ad una Lista ordinata in modo crescente ottenuta fondendo le liste  $l1$  e  $l2$ . La procedura *merge* deve essere ricorsiva e non puo' utilizzare allocazione dinamica della memoria: puo' invece distruggere le liste  $l1$  e  $l2$ .

### ESERCIZIO 5

Riscrivere il seguente frammento di codice senza utilizzare il comando `switch`.

```
switch (x+y)
{
case 0: case 8: case 16: z=2;
case 2: case 4: z=z+1; break;
case 21: z=z-1;
default : z = 2*z;
}
```

### ESERCIZIO 6

Supponendo date le seguenti dichiarazioni

```
int x;
char c;
double d;
```

indicare il tipo delle seguenti espressioni:

(1) `c + d`                    (2) `(int) d + (float) (x+1)`  
(3) `c + x + 1.0`                (4) `x/c`

### ESERCIZIO 7

Indicare lo stato della pila nel punto indicato (cioè immediatamente prima dell'esecuzione del comando `return`) in corrispondenza delle due chiamate della funzione `f`.

```
int f(int x, int y)
{
    y = y+x;
    /* PILA A QUESTO PUNTO */
    return (y+1);
}
main()
{
    int x = 10, z = 20;
    x = f(f(x,z),x);
}
```