

INFORMATICA

MATRICOLA: COGNOME: NOME:

CARRAZZA

MEREGHETTI

TAMASCELLI

ESERCIZIO 1. Un array X di d interi si dice *limitato* se e solo se soddisfa entrambe le seguenti condizioni:

(1) contiene solo interi *non negativi*,

(2) per ogni posizione $i \in \{0, \dots, d-1\}$, vale che $\sum_{k=0}^i X[k] \leq i$.

Ad esempio, l'array $X = \{0, 1, 0, 2, 0, 1, 1, 2\}$ è limitato.

- Scrivete la funzione `bool bounded(int *X, int d)` che ritorna `true` se l'array X di dimensione d è *limitato*, `false` altrimenti.

```
bool bounded( int *X, int d ) {
    int somma = 0;
    int conte = 0;
    for (int i = 0; i < d; i++) {
        somma = somma + X[i];
        if (somma <= i) {
            conte++;
        }
    }
    if (conte == d) return true;
    else return false;
}
```

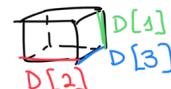
ESERCIZIO 2. Una *scatola* può essere memorizzata in variabili di tipo

```
struct box { char c; float D[3]; }
```

in cui il vettore D contiene ordinatamente *altezza*, *larghezza* e *profondità* della scatola mentre il campo c ne contiene un *codice* rappresentato da un singolo carattere.

- Scrivete la funzione `float supvol(box a, box b)` che restituisce la somma dei *volumi* delle scatole a e b se esse hanno il medesimo codice, altrimenti restituisce la somma delle aree delle *superficie* di a e di b .

```
float supvol (box a, box b) {
    float V;
    float A;
    V = (a.D[1] * a.D[2] * a.D[3]) + (b.D[1] * b.D[2] * b.D[3])
    A = ((a.D[1] + a.D[3]) * 2 * a.D[2]) + ((b.D[1] + b.D[3]) * 2 * b.D[2])
    if (a.c == b.c) return V;
    else return A;
}
```



ESERCIZIO 3. Il file `numbers` contiene un numero impreciso di interi positivi, uno per riga.

- Scrivete un frammento di codice che stampi: (i) la media degli interi pari e (ii) la media degli interi dispari contenuti nel file `numbers` senza memorizzare gli interi in un vettore.

```
int npari = 0;
int ndisp = 0;
int somp = 0;
int somd = 0;

ifstream filein;
filein.open("numbers");
if (filein.fail()) return -1;
int appo;
filein >> appo;
while (filein.eof() == false){
    if (appo % 2 == 0){
        npari++;
        somp = somp + appo;
    }
    else {
        ndisp++;
        somd = somd + appo;
    }
    filein >> appo;
}

cout << "media pari" << (float)somp / npari << endl;
cout << "media dispari" << (float)somd / ndisp << endl;
```

ESERCIZIO 4. La seguente funzione *ricorsiva* può essere richiamata ESCLUSIVAMENTE su interi positivi:

```
int funz( int a, int b ) {
    if ( a < b )
        return a;
    else
        return funz( a - b, b );
}
```

- Indicare quale valore intero restituisce la chiamata `funz(155, 34)`:
- In generale, che significato ha il valore intero restituito dalla chiamata `funz(x, y)` con x, y interi positivi?
.....