PROBLEM SOLVING SU ARRAY

LO GUARDATE ANCHE GLI APPUNTI DI BARBUTI (CORSO A) DEL 2/10

```
int member(int a[], int dim, int x) {
    int i=0;
    int Trovato = 0;
    while (i<dim && !Trovato) {
        if (a[i] == x) Trovato=1;
        i++;
    }
    return Trovato;
}
```

UNA VARIABILE USATA IN QUESTO MODO VIENE DETTA "FLAG"

RISPONDE A QUESTA DOMANDA:

$$\exists\, i \in [0, dim) \,.\, a[i] == x \quad ?$$

ESISTE UN INDICE i COMPRESO TRA 0 E DIM (QUEST'ULTIMO ESCLUSO) TALE CHE a[i] È UGUALE A x ?

$[0, dim)$

RAPPRESENTA L'INTERVALLO CHE COMPRENDE I VALORI DA 0 A dim

↑ COMPRESO ↑ ESCLUSO

COMPRESO ESCLUSO DA 0 A dim

0 dim

CORRISPONDE A $[0, dim-1]$

ALCUNI ESEMPI:

- $\exists i \in [0, dim) \,.\, a[i] < 0$ — ESISTE UN ELEMENTO NEGATIVO?

- $\exists i \in [0, dim) \,.\, a[i] \,\%\, 2 == 0$ — ESISTE UN NUMERO PARI?

- $\exists i \in [0, dim-1) \,.\, a[i] == a[i+1]$ — ESISTONO DUE ELEMENTI CONSECUTIVI UGUALI?

L'ULTIMO ELEMENTO NON HA UN SUCCESSORE!!

IN GENERALE

$\exists i \in [0, dim) \,.\, P$ ← PROPRIETÀ DELL'ELEMENTO CHE STO CERCANDO

PUÒ VARIARE

RICERCA DI UN NUMERO NEGATIVO

$\exists i \in [0, dim) \cdot a[i] < 0$

```
int negativo (int a[], int dim) {
    int i=0;
    int Trovato=0;
    while (i<dim && !Trovato) {
        if (a[i]<0) Trovato=1;
        i++;
    }
    return Trovato;
}
```

ESTREMI DELL'INTERVALLO

CONSIDERIAMO ORA QUESTI:

- $\forall i \in [0, dim) . a[i] < 0$ SONO TUTTI NEGATIVI ?
- $\forall i \in [0, dim) . a[i] \% 2 == 0$ SONO TUTTI PARI ?
- $\forall i \in [0, dim-1) . a[i] == a[i+1]$ SONO TUTTI UGUALI ?

$\Longleftrightarrow$

$\forall i \in [1, dim) . a[i] == a[0]$

TIPICA SOLUZIONE SBAGLIATA

```
int TuttiPari (int a[], int dim) {
    int pari = 0;
    for (int i=0; i<dim; i++) {
        if (a[i]%2==0) pari = 1;
        else pari = 0;
    }
    return pari;
}
```

a | 2 | 4 | 6 | 5 |
0 1 2 3
P P P P

pari 0 ~~1~~ ~~1~~ ~~1~~ 0

OK

FATE

FATE LE SIMULAZIONI SU CARTA DEI VOSTRI PROGRAMMI!!

a | 2 | 5 | 7 | 4 |
0 1 2 3

pari ∅

ok

1 ∅ ∅ 1

NO

IDEA:

SONO TUTTI PARI

SE E SOLO SE

NESSUNO È DISPARI

IN TERMINI LOGICI:

$\forall i \in [0, dim) \,.\, a[i] \,\%\, 2 == 0$ VERA

SE E SOLO SE

$\nexists i \in [0, dim) \,.\, a[i] \,\%\, 2 == 1$ FALSA

QUESTO LO SAPPIAMO FARE, SOLO CHE DEVO DARE IL RISULTATO CONTRARIO (FALSO)

```
int TuttiPari (int a[], int dim) {
    int i=0;
    int Trovato=0;
    while (i<n && !Trovato) {
        if (a[i]%2==1) Trovato=1;
        i++;
    }
    return !Trovato;
}
```

ALTRO MODO EQUIVALENTE:

```
int TuttiPari (int a[], int dim) {
    int i=0
    int ok=1;
    while (i<dim && ok) {
        if (a[i]%2==1) ok=0;
        i++
    }
    return ok;
}
```

NON C'È IL !

PROBLEMI DI CONTEGGIO

→ CONTARE QUANTI NUMERI PARI CI SONO IN UN ARRAY

AND LOGICO

$$\#\{ i \mid i \in [0, dim) \wedge a[i] \%\, 2 == 0 \}$$

INSIEME DEGLI INDICI i CUI ELEMENTI CORRISPONDENTI SONO PARI

CARDINALITÀ DELL'INSIEME (OSSIA IL NUMERO DI ELEMENTI CHE CONTIENE)

a

8	7	3	2
0	1	2	3

$\{0, 3\}$

(INSIEME DEGLI INDICI DI ELEMENTI PARI

$\#\{0,3\} = 2$

$\#\{5,7,9\} = 3$

⋮

```
int contapari (int a[], int dim) {
    int conta = 0;
    for (int i=0; i<dim; i++) {
        if (a[i]%2==0) conta++;
    }
    return conta;
}
```

UNA VARIABILE USATA IN QUESTO MODO È DETTA "CONTATORE"

a | 4 | 5 | 2 | 7 |
0 1 2 3

conta 0 1 2

→ PROBLEMI DI AGGREGAZIONE

→ CALCOLARE LA SOMMA DI TUTTI I PARI

$$\sum_{i \in [0,dim)\ a[i]\%2==0} a[i]$$

SOMMATORIA

FACCIO LA SOMMA DI TUTTI GLI ELEMENTI $a[i]$ TALI CHE $i \in [0, dim)$ E $a[i]\%2==0$

```
int sommaPositivi (int a[], int dim) {
    int somma = 0
    for (int i=0; i<dim; i++) {
        if (a[i]%2==0) somma += a[i];
    }
    return somma;
}
```

UNA VARIABILE USATA IN QUESTO MODO È DETTA "ACCUMULATORE"

`x+=y;` ≡ `x=x+y;`

a | 6 | 5 | 2 | 1 |
0 1 2 3

Somma | ∅ | → 8

RICERCA DEL MINIMO ELEMENTO

→ SCRIVERE UNA FUNZIONE CHE RESTITUISCE UN VALORE X TALE CHE

$$\exists i \in [0, dim) . \left(a[i] == x \wedge \forall j \in [0, dim) . x <= a[j] \right)$$

X È PRESENTE E X È MINORE O UGUALE DI TUTTI GLI ELEMENTI

PARTIAMO SCRIVENDO UNA FUNZIONE CHE CALCOLA

$$\forall j \in [0, dim) . x <= a[j]$$

CORRISPONDE A VERIFICARE

$$\exists j \in [0, dim) . x > a[j] \text{ È FALSO}$$

```
int minoreDiTutti (int a[], int dim, int x) {
    int i=0;
    int ok=1;
    while ( i<dim && ok ) {
        if (x>a[i]) ok=0;
        i++;
    }
    return ok;
}
```

ORA CHE HO LA FUNZIONE MINOREDITUTTI

VERIFICARE

$$\exists i \in [0, dim) . \left(a[i] == x \wedge \forall j \in (0, dim) . x <= a[j] \right)$$

CORRISPONDE A FARE

$$\exists i \in [0, dim) . \left(a[i] == x \wedge minoreditutti(a, dim, a[i]) \right)$$

DEVE RESTITUIRE → X

```
int minimo (int a[], int dim) {
    int i = 0;
    int Trovato = 0;
    int min;
    while ( i < dim && !Trovato) {
        if (minoreditutti (a, dim, a[i])) {
            min = a[i];
            Trovato = 1;
        }
    }
    return min;
}
```

POSSIAMO RISOLVERE PROBLEMI COMPLESSI COMPONENDO (MODULARMENTE) SOLUZIONI DI

PROBLEMI PIU' SEMPLICI

RAGIONANDO SUL PROBLEMA MI RENDO CONTO CHE POSSO TROVARE UNA SOLUZIONE PIU' FURBA CHE SI "PORTA DIETRO" NELLO SCANDIRE L'ARRAY IL VALORE MINIMO TROVATO FINO A QUEL MOMENTO

```
int minimo (int a[], int dim)
    int min = a[0];
    for (int i = 1; i<dim; i++) {
        if (a[i] < min)
            min = a[i];
    }
    return min;
}
```

min RAPPRESENTA IL VALORE MINIMO CHE HO INCONTRATO FINO A QUEL MOMENTO

STO ASSUMENDO CHE ESISTA ALMENO UN ELEMENTO (dim > 0)

DOMANDE PRATICHE:

→ COME FACCIO A FARE ESERCIZI ?

↳ VEDREMO ALCUNI ESERCIZI DA SVOLGERE SU CARTA

↳ IN LABORATORIO

→ COME SI SCRIVONO LE FUNZIONI ?

↳ LE FUNZIONI DEVONO ESSERE DEFINITE (SCRITTE) PRIMA DELLA LORO CHIAMATA

```
int somma (int x, int y) {
    ⋮
}

main() {
    ⋮
    somma(3,2);
    ⋮
}
```

SE INVERTO IL COMPILATORE C MI DA UN ERRORE

OK

POTREI SCRIVERE SOMMA DOPO IL MAIN A PATTO DI SCRIVERE IL PROTOTIPO DI SOMMA

```
somma (int x, int y);
```

PRIMA DEL MAIN