

Programmazione in C (ANSI 89)

Analisi soluzioni

Esercizio 1 - appello 05/02/2021

Scrivere un sottoprogramma baricentro che ricevuto in ingresso un array di valori interi e qualsiasi altro parametro ritenuto strettamente necessario restituisce l'indice dell'elemento che fa da baricentro agli elementi dell'array, -1 se non esiste. Definiamo baricentro di un array di h elementi, l'elemento in posizione i tale per cui: $v[0] + v[1] + \dots + v[i] = v[i+1] + v[i+2] + \dots + v[h-1]$

Se dovessero esserci più baricentri, il sottoprogramma restituirebbe quello più a sinistra.

ESEMPIO:

in: 1 0 5 10 4 4 2 6 0 0

out: 3

La somma dei primi 4 elementi è uguale a 16 come quella degli ultimi 6 elementi

in: 1 0 5 10 4 4 2 7 0 0

out: -1

Esercizio 2 - appello 20/01/2020

Scrivere un sottoprogramma che riceve in ingresso un array di valori interi v e qualsiasi altro parametro ritenuto necessario, ed altri due valori interi da e a . Il sottoprogramma verifica se nell'array sono presenti tutti e soli i valori inclusi nell'intervallo $[da, a]$, senza ripetizioni. In caso positivo il sottoprogramma restituisce 1, 0 in caso contrario.

Esercizio 3 - appello 26/01/2022

Scrivere un sottoprogramma che ricevuto in ingresso un valore intero restituisce al chiamante il numero di cifre che compaiono più di una volta. Per esempio, se il sottoprogramma riceve in ingresso il valore 3282821 il sottoprogramma restituisce il valore 2, perché la cifra 8 e 2 compaiono entrambe più di una volta.

Esercizio 4 - appello 20/01/2020

Scrivere un sottoprogramma `minmaxstr` che riceve in ingresso una stringa e trasmette al chiamante due caratteri, il minimo e il massimo nell'ordinamento alfabetico tra quelli contenuti nella stringa. Per esempio, se la stringa in ingresso è "esempio", il minimo ed il massimo sono rispettivamente 'e' ed 's'. La stringa contenga tutti e soli caratteri alfabetici minuscoli.

Rivedere il sottoprogramma `minmaxstr` in modo tale che i caratteri possano essere sia maiuscoli, sia minuscoli, e facendo in modo che il sottoprogramma trasmetta comunque quelli minuscoli. In tal caso, se la stringa in ingresso è "Architetto", il minimo ed il massimo sono rispettivamente 'a' ed 't'.

Esercizio 5 - appello 17/02/2020

In un array bidimensionale di valori interi, si definisce dominante ogni elemento dell'array che è strettamente maggiore di tutti gli elementi dell'array bidimensionale che si trova in basso a destra rispetto all'elemento stesso (si veda la figura), non considerando però tutti gli elementi presenti nell'ultima colonna e nell'ultima riga. Si realizzi un sottoprogramma che ricevuto in ingresso un array bidimensionale e qualsiasi altro parametro ritenuto strettamente necessario calcoli e restituisca al chiamante il numero di elementi dominanti presenti. Nel contesto di utilizzo del sottoprogramma, sono presenti le seguenti direttive/istruzioni riportate di seguito.

```
#define NR ...
#define NC ...
...
int main(int argc, char * argv[])
{
    ...
    int info[NR][NC];
}
```

5	9	2	4	1	7	2	4
3	5	6	2	5	6	1	2
1	3	4	7	8	8	3	0
1	3	5	6	7	8	2	1

(a) elemento dominante e array dominato

5	9	2	4	1	7	2	4
3	5	6	2	5	6	1	2
1	3	4	7	8	8	3	0
1	3	5	6	7	8	2	1

(b) elementi dominanti

Esercizio 6A - appello 01/09/2021

Una applicazione richiede di scegliere il proprio nome utente nel rispetto di alcune regole e si vuole realizzare un programma che acquisisca e verifichi il nome scelto.

Scrivere il sottoprogramma valido che ricevuti in ingresso il nome scelto (username), due interi (min e max) e una stringa (speciali) verifica che esso rispetti le seguenti regole:

- + deve essere di almeno min caratteri,
- + non deve superare la lunghezza di max caratteri,
- + deve contenere almeno un carattere appartenente ai caratteri presenti nella stringa speciali.

Nel caso in cui tutte le regole siano rispettate, il nome utente scelto è valido e il sottoprogramma restituisce 1, in caso contrario il sottoprogramma restituisce 0.

Nota: non effettuare elaborazioni inutili e/o inefficienti: la soluzione proposta verrà valutata anche in relazione a questo aspetto.

Esercizio 6B - appello 01/09/2021

Scrivere un programma che acquisisce dall'utente una stringa di al più 20 caratteri e verifica che questa contenga un nome utente valido (avvalendosi del sottoprogramma valido), e fino a che non è tale continua a richiederla. Quando le regole sono rispettate il programma chiama il sottoprogramma StartApp che riceve in ingresso il nome utente (il prototipo è riportato di seguito) e realizza la funzionalità desiderata.

```
int StartApp(char []);
```

I vincoli per la validità del nome utente sono una lunghezza compresa nell'intervallo [8; 15] e un insieme dei caratteri speciali costituito dai caratteri '\$' '#' '%' '&'.