

Laboratorio 6

Nella propria home directory creare una sottodirectory chiamata es06, in cui metteremo tutti i file C di oggi.

Esercizi su procedure e funzioni

Nel seguito, quando si dice "riceve un array" o qualcosa di equivalente, si intende ***SEMPRE*** che venga passata anche la lunghezza dell'array come parametro, anche se non viene esplicitamente detto.

Inoltre si richiede che, oltre alla funzione o procedura che e' stata richiesta, scriviate anche un main che usi tale funzione e ne dimostri ***ogni*** funzionalità'.

Non e' invece necessario che scriviate le funzioni in file separati, potete lavorare per ogni esercizio su un unico file.

Per ultimo, se con l'esercizio viene fornita la firma(o prototipo) della funzione/procedura da scrivere, si richiede che la soluzione rispetti questa firma in ogni dettaglio

(nome funzione, tipo di ritorno, lista di argomenti).

Ricordate che una procedura e' una funzione con tipo di ritorno void.

Per quanto riguarda l'uso dei booleani utilizzare

```
#DEFINE FALSE 0
```

```
#DEFINE TRUE 1
```

come visto a lezione.

1) Scrivere un programma che dichiari un array di caratteri di dimensione DIM (dove DIM e' una costante definita con un DEFINE), lo inizializzi con DIM caratteri letti dall'input e poi stampi il contenuto dell'array a video.

2) Scrivere una funzione che riceva un array di interi e ritorni l'ultimo elemento dell'array.

3) Scrivere una procedura che, ricevuto un array di interi, ritorni la somma di tutti gli elementi e azzeri tutti gli elementi dispari.

4) Scrivere una funzione che, ricevuto un array di caratteri, ritorni la prima lettera maiuscola, se c'e'. Se non c'e', allora ritorna '\n'.

5) Scrivere una procedura che ricevuto un array di caratteri, trasformi ogni lettera minuscola in maiuscola, lasciando inalterati gli altri caratteri

6) Dato un array di interi, scrivere una funzione che controlli che l'array contenga solo elementi pari.

7) Scrivere una funzione che ricevuto un array di interi, ritorni la somma degli elementi in posizione dispari meno la somma degli elementi in posizione pari.

8) Dato un array di interi non vuoto, scrivere una funzione che ritorni il numero di occorrenze del valore minimo (scorrendo l'array una volta sola).

9) Scrivere una funzione/procedura che ricevuto un array di interi, restituisca la posizione della prima e dell'ultima occorrenza di un elemento pari nell'array.

10) Scrivere una funzione che verifichi che un vettore sia ordinato in senso decrescente.

11) Scrivere una funzione che riceva una matrice bidimensionale di interi e azzeri ogni elemento il cui contenuto è maggiore o uguale alla somma delle sue coordinate.
void azzera(int mat[][5], int numRighe);

12) Esercizio 5: MSS, Maximum Segment Sum

Dato un array di interi positivi e negativi, il segmento di somma massima è la porzione contigua dell'array in cui la somma degli elementi ha valore più alto. Ad esempio l'array

[2,-4,2,-1,6-3]

ha come SSM il segmento [2,-1,6] di valore 7. Si chiede di definire due funzioni per la stampa e per il calcolo di SSM, con i seguenti prototipi :

```
/** stampa l'array s di lunghezza n */  
void print_array(int s[], int n);
```

```
/** calcola SSM sull'array s di lunghezza lung  
\param s array  
\param n lunghezza  
\param s_init puntatore alla variabile che conterra' la posizione di inizio dell'SSM  
\param s_lung puntatore alla variabile che conterra' la lunghezza del segmento di  
somma massima
```

```
\retval k la somma degli elementi dell' SSM  
*/  
int ssm (int s[], int n, int * s_init, int * s_lung);
```