Puntatori

Informatica A - 24/10/2024

Esempio - Calcolo del cubo

```c
#include <stdio.h>

int cubeByValue(int n); // prototipo

int main(void) {
   int number = 5; // inizializza number

   printf("The original value of number is %d", number);
   number = cubeByValue(number); // passa number per valore a cubeByValue
   printf("\nThe new value of number is %d\n", number);
}

// calcola e restituisci il cubo di un argomento intero
int cubeByValue(int n) {
   return n * n * n; // restituisci il cubo di n
}
```

```
The original value of number is 5
The new value of number is 125
```

```c
#include <stdio.h>

void cubeByReference( int *nPtr); // prototipo di funzione

int main(void) {
   int number = 5; // inizializza number

   printf("The original value of number is %d", number);
   cubeByReference(&number); // passa l'indirizzo di number a cubeByReference
   printf("\nThe new value of number is %d\n", number);
}

// eleva al cubo *nPtr; di fatto modifica number in main
void cubeByReference( int *nPtr ) {
   *nPtr = *nPtr * *nPtr * *nPtr; // calcola il cubo di *nPtr
}
```

```
The original value of number is 5
The new value of number is 125
```

Esercizio 1

Per ciascuna delle seguenti richieste, scrivi una singola linea di codice che faccia quanto indicato. (Assumi che value1 e value2 siano due long int e che value1 sia stato inizializzato a 200000)

a) Definisci la variabile lPtr come puntatore di un oggetto di tipo long
b) Assegna l'indirizzo di value1 al puntaote lPtr
c) Print il valore dell'oggetto cui lPtr punta
d) Assegna il valore dell'oggetto cui lPtr punta a value2
e) Print il valore di value2
f) Print l'indirizzo di value1
g) Print l'indirizzo racchiuso in lPtr. E' lo stesso valore di value 1?

Esercizio 2

Ci sono errori nel seguente frammento di codice? Cosa fa?

```c
#include <stdio.h>
#define SIZE 80

void mystery1(char *s1, const char *s2); // prototype

int main(void) {
   char string1[SIZE]; // create char array
   char string2[SIZE]; // create char array

   puts("Enter two strings: ");
   scanf("%39s%39s" , string1, string2);
   mystery1(string1, string2);
   printf("%s", string1);
}

// What does this function do?
void mystery1(char *s1, const char *s2) {
   while (*s1 != ) {
      ++s1;
   }

   for (; *s1 = *s2; ++s1, ++s2) {
      ; // empty statement
   }
}
```

Esercizio 3

Ci sono errori nel seguente frammento di codice? Cosa fa?

```c
#include <stdio.h>
#define SIZE 80

size_t mystery2(const char *s); // prototype

int main(void) {
   char string[SIZE]; // create char array

   puts("Enter a string: ");
   scanf("%79s", string);
   printf("%d\n", mystery2(string));
}

// What does this function do?
size_t mystery2(const char *s) {
   size_t x;

   // loop through string
   for (x = 0; *s != ; ++s) {
      ++x;
   }

   return x;
}
```

Esercizio 4

Ci sono errori nel seguente frammento di codice? Cosa fa?

```c
#include <stdio.h>
#define SIZE 80

int mystery3(const char *s1, const char *s2); // prototype

int main(void) {
   char string1[SIZE]; // create char array
   char string2[SIZE]; // create char array

   puts("Enter two strings: ");
   scanf("%79s%79s", string1 , string2);
   printf("The result is %d\n", mystery3(string1, string2));
}
int mystery3(const char *s1, const char *s2) {
   int result = 1;

   for (; *s1 !=  && *s2 != ; ++s1, ++s2) {
      if (*s1 != *s2) {
         result = 0;
      }
   }

   return result;
}
```

Esercizio 5

Scrivere un programma C che definita una matrice di array, per ciascuno ne mostri a video la media.

Esercizio 6

Scrivere una funzione per la risoluzione di equazioni di secondo grado della forma $ax2 + bx + c = 0$ in base alle seguenti indicazioni. La funzione riceve tre parametri (di tipo double) passati per valore che rappresentano i tre coefficienti a, b e c e due parametri x1 e x2 (anch'essi di tipo double) passati per riferimento. La funzione ritorna un valore intero. Se l'equazione ha soluzioni reali la funzione ritorna 1 e nei parametri x1 e x2 vengono scritte le soluzioni dell'equazione. Se l'equazione non ha soluzioni reali la funzione ritorna 0 e le variabili x1 e x2 non vengono scritte.

Scrivere poi un programma che fa inserire all'utente i coefficienti di un'equazione di secondo grado e stampa le sue soluzioni se queste sono reali e un opportuno messaggio in caso contrario.

Esercizio 7

Modificare la funzione precedente in modo che calcoli le soluzioni complesse dell'equazione. Più precisamente, la funzione avrà tre parametri (di tipo double) passati per valore che rappresentano i coefficienti a, b e c dell'equazione e quattro parametri (di tipo double) passati per riferimento. Tali parametri rappresentano la parte reale e la parte immaginaria di ciascuna delle due soluzioni.

Esercizio 8

Scrivere una funzione che calcola l'equazione di una retta nel piano passante per due punti (x1,y1) e (x2, y2). La funzione riceve come parametri i quattro valori double x1, y1, x2 e y2 (passati per valore) e due ulteriori parametri *m* e *q* (passati per riferimento) in cui la funzione deve scrivere il coefficiente angolare *m* e il termine noto *q* dell' equazione della retta passante per i due punti dati. Qualora la retta sia verticale (cioè avente equazione y=c) il valore di *m* deve essere impostato a INFINITY (tale costante è definita nella libreria math.h) e *q* deve essere uguale a c.

Scrivere poi un programma che chiede all'utente di inserire le coordinate di due punti nel piano e stampa l'equazione della retta passante per i due punti dati.

Esercizio 9

Scrivere una funzione che prende come parametro un array di interi e restituisce il numero di elementi che sono maggiori del loro successore. Si operi sugli elementi dell'array tramite l'aritmetica dei puntatori invece che tramite gli indici.

Esercizio 10

Scrivere un programma C che definita una matrice 4x4, estragga e stampi a video la sottomatrice 2x2 la cui somma dei valori è massima.

Esercizio 11

Si scriva una funzione ricorsiva che prende in input due array della stessa dimensione A e B, i cui elementi sono solo numeri 0 ed 1 e calcola la loro distanza di Manhattan.

La distanza di Manhattan è definita come il numero di elementi diversi nei due array.

(Ad esempio, gli array: 1,0,1,1,0,1 e 0,1,1,1,0,1 Hanno distanza di Manhattan uguale a 2)

Esercizio 12

Scrivere un programma C che riceva in input un numero n di stringhe e le renda tutte maiuscole.