

Esercizio 4 assegnato il 30 gennaio 2014.

Da consegnare corretto entro il 3/2 compreso.

Viene dato un main che apre i soliti file "input" e "output", dichiara un array `int A[3][3][4][5]`, legge 180 valori da "input" in A (nell'ordine in cui gli elementi di A sono allocati nella memoria) e poi invoca una funzione

```
void torta_max(int X[][3][4][5], int n_torte, int* p_max)
```

che deve soddisfare la seguente coppia di pre- e post-condizioni:

PRE=(`n_torte > 0`, `X[n_torte][3][4][5]` è completamente definito, `p_max` punta ad una variabile `max` del main)

POST=(la funzione inserisce in `max` l'indice della torta di A che contiene lo strato a somma massima)

Spiegazione aggiuntiva: ci sono 3 torte da considerare (`torta_max` usa il parametro formale `n_torte` anziché 3 perché in questo modo funzionerebbe anche con numero di torte diverso da 3). Per ciascuna torta si deve calcolare qual è il massimo tra le somme degli elementi di ciascuno strato. Consideriamo la prima torta e supponiamo che il suo primo strato abbia somma degli elementi pari a 100, il secondo strato abbia somma pari a 90 e il terzo abbia somma pari a 120. Quello che dobbiamo ricordare è 120 (il massimo tra i valori) e poi dobbiamo fare lo stesso calcolo per la seconda torta (supponiamo che la somma massima dei suoi strati sia -10 (negativo, perché no?)) e infine supponiamo che per la terza torta la somma massima degli strati sia 231. Ovviamente la torta con strato a somma massima è la terza, cioè quella con indice 2 e quindi, come richiesto dalla POST, `torta_max`, tramite il parametro formale `p_max`, dovrà assegnare 2 alla variabile `max` del main che alla fine del main viene stampata su "output".

Il main è come segue:

```
main()
```

```
{
  ifstream INP("input");
  ofstream OUT("output");
  if(INP && OUT)
  {
    int A[3][3][4][5], max, *p=***A;

    for(int i=0; i<3*3*4*5; i++)
      INP>>p[i];

    torta_max(A,3, &max); //questa è l'invocazione alla funzione da fare

    OUT<<max<<endl;
    INP.close();
    OUT.close();
  }
}
```

```
else
    cout<<"errore nei file"<<endl;
}
```

Correttezza: ogni ciclo (while o for) deve avere un associato invariante ed una post-condizione. Per almeno un ciclo è richiesta una dimostrazione di correttezza (che comprende le 3 parti richieste per tali prove).

Notare: la funzione `torta_max` può invocare anche un'altra funzione che a sua volta ne può invocare una terza. In caso vengano introdotte funzioni, oltre a `torta_max`, ciascuna di esse deve avere la sua coppia di pre- e post-condizioni.