|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COGNOME:** |  | **NOME** |  | **A1** |
| **MATRICOLA:** |  | | |
| **DOCENTE:** |  | | |

|  |
| --- |
| **Domanda 1** |
| Considerando i due numeri n1=01010101 e n2=11110101 rappresentati in CA2. Si esegua la somma indicando opportunamente la presenza di overflow (i numeri sono espressi su 8 bit). |
|  |

|  |
| --- |
| **Domanda 2** |
| Una memoria è in grado di memorizzare 10KB. Considerando un parallelismo dati di 8 bit, si calcoli quanti bit sono necessari per gli indirizzi. |
|  |

|  |
| --- |
| **Domanda 3** |
| Si descriva l’architettura di un BUS e si illustri il suo principio di funzionamento |
|  |

|  |
| --- |
| **Domanda 4 (PROGRAMMAZIONE)** |

Si desidera realizzare un software in grado di elaborare le immagini ottenute attraverso i DNA microarray. Un DNA microarray è un insieme di microscopiche sonde di DNA attaccate ad una superficie solida come un chip di silicio formanti una **matrice rettangolare**. La dimensione della matrice è NxM di numeri interi positivi variabili tra 0 e 200 e separati da uno spazio.

Ogni numero rappresenta il livello di espressione di un gene. Un gene è considerato i**dentificato** quando il suo valore supera il 70% del valore massimo (200), è **non identificato** quando il suo valore è inferiore al 30%. Nel caso intermedio è **parzialmente identificato**.

Si scriva un programma in linguaggio C in grado di:

* analizzare una matrice **5x10** rispetto ad un sottoinsieme di **coordinate**
* verificare il **tipo di identificazione** del sottoinsieme di geni.

Tale sottoinsieme è contenuto in un secondo file passato da linea di comando ed avente il seguente formato:

**<riga> <colonna> <tipo>**

Riga e colonna sono due numeri interi, mentre tipo è un carattere che può assumere i seguenti valori: **N** (non identificato), **E** (identificato), **P** (parzialmente identificato). Il numero di coordinate da verificare non è noto a priori.

Il programma dovrà:

1. Stabilire se tutti i codici identificativi dei geni appartengono alla matrice.
   1. In caso negativo, stampare a video i codici non trovati e terminare.
   2. In caso positivo procedere al punto 2.
2. Verificare se tutti i codici identificativi rispettano il tipo di identificazione
   1. In caso positivo, stampare a video un messaggio di conferma
   2. In caso negativo, il programma dovrà stampare a video la matrice riportando il valore 0 per i geni che rispettano il tipo di identificazione oppure non sono definiti e un valore numerico corrispondente al livello da sommare per raggiungere il criterio di identificazione del gene (il valore può essere negativo).

Esempi di file:

|  |  |  |  |
| --- | --- | --- | --- |
| DNA\_image.txt  3 195 193 88 50 2 4 199 198 7  4 194 192 85 2 8 10 4 5 8  80 81 10 5 5 6 5 9 10 70  10 4 58 70 80 70 80 81 91 6  190 189 5 6 7 10 11 9 43 40 | Model\_1.txt  11 1 E  1 9 E  4 6 N  10 10 P  4 7 E | Model\_2.txt  0 1 E  0 9 N  2 0 P  4 1 E | Model\_3.txt  0 3 N  2 9 N  3 2 P  0 8 E |

C:\prog.exe DNA\_image.txt Model\_1.txt

Geni identificativi non trovati:

11 1 E

10 10 P

C:\prog.exe DNA\_image.txt Model\_2.txt

Geni identificativi correttamente espressi.

C:\prog.exe DNA\_image.txt Model\_3.txt

0 0 0 -28 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 -10

0 0 2 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COGNOME:** |  | **NOME** |  | **A2** |
| **MATRICOLA:** |  | | |
| **DOCENTE:** |  | | |

|  |
| --- |
| **Domanda 1** |
| Considerando i due numeri n1=10000001 e n2=11111111 rappresentati in CA2. Si esegua la somma indicando opportunamente la presenza di overflow (i numeri sono espressi su 8 bit). |
|  |

|  |
| --- |
| **Domanda 2** |
| Una memoria è in grado di memorizzare 20KB. Considerando un parallelismo dati di 16 bit, si calcoli quanti bit sono necessari per gli indirizzi. |
|  |

|  |
| --- |
| **Domanda 3** |
| Si descrivano le caratteristiche di una memoria RAM. |
|  |

|  |
| --- |
| **Domanda 4 (PROGRAMMAZIONE)** |

Si desidera realizzare un software in grado di elaborare le immagini ottenute attraverso i DNA microarray. Un DNA microarray è un insieme di microscopiche sonde di DNA attaccate ad una superficie solida come un chip di silicio formanti una **matrice rettangolare**. La dimensione della matrice è NxM di numeri interi positivi variabili tra 0 e 200 e separati da uno spazio.

Ogni numero rappresenta il livello di espressione di un gene. Un gene è considerato i**dentificato** quando il suo valore supera il 70% del valore massimo (200), è **non identificato** quando il suo valore è inferiore al 30%. Nel caso intermedio è **parzialmente identificato**.

Si scriva un programma in linguaggio C in grado di:

* analizzare una matrice **5x10** rispetto ad un sottoinsieme di **coordinate**
* verificare il **tipo di identificazione** del sottoinsieme di geni.

Tale sottoinsieme è contenuto in un secondo file passato da linea di comando ed avente il seguente formato:

**<riga> <colonna> <tipo>**

Riga e colonna sono due numeri interi, mentre tipo è un carattere che può assumere i seguenti valori: **N** (non identificato), **E** (identificato), **P** (parzialmente identificato). Il numero di coordinate da verificare non è noto a priori.

Il programma dovrà:

1. Stabilire se tutti i codici identificativi dei geni appartengono alla matrice.
   1. In caso negativo, stampare a video il numero di geni non trovati.
   2. In caso positivo procedere al punto 2.
2. Verificare se tutti i codici identificativi rispettano il tipo di identificazione
   1. In caso positivo, stampare a video un messaggio di conferma
   2. In caso negativo, il programma dovrà stampare a video la sequenza di geni seguiti da OK (se il valore identificativo è corretto) o dal valore da sommare per raggiungere il criterio di identificazione riportato nel secondo file (il valore può essere negativo).

Esempi di file:

|  |  |  |  |
| --- | --- | --- | --- |
| DNA\_image.txt  3 195 193 88 50 2 4 199 198 7  4 194 192 85 2 8 10 4 5 8  80 81 10 5 5 6 5 9 10 70  10 4 122 70 80 70 80 81 91 6  190 189 5 6 7 10 11 9 43 40 | Model\_1.txt  11 1 E  1 9 E  4 6 N  10 10 P  4 7 E | Model\_2.txt  0 1 E  0 9 N  2 0 P  4 1 E | Model\_3.txt  0 3 N  2 9 N  3 2 P  0 4 E |

C:\prog.exe DNA\_image.txt Model\_1.txt

Geni identificativi non trovati: 2

C:\prog.exe DNA\_image.txt Model\_2.txt

Geni identificativi correttamente espressi.

C:\prog.exe DNA\_image.txt Model\_3.txt

0 3 OK

2 9 OK

3 2 -2

0 4 70