|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CORSO** | **Informatica – 12BHD** | | | | **DATA** | | **01/09/2014** | | | | |
| **NOME** |  | | | | **COGNOME** | |  | | | | |
| **MATRICOLA** | | **S** |  |  | |  | |  |  |  | **C/1** |
| ☐AAA-BARC ☐BARD-BOUH ☐BOUI-CART ☐CARU-CONS ☐CARU-CONS ☐CONT-DEMAR ☐DEMAS-FERRD ☐FERRE-GIAQ ☐GIAR-LAEZ ☐LAFA-MANC ☐MAND-MIQZ ☐MIRA-PAHZ  ☐PAIA-PODD ☐PODE-ROSSE ☐ROSSF-SIQZ ☐SIRA-TUCB ☐TUCC-ZZZ  ☐Poli@Home ☐5 Crediti ☐AAA-LIB/English ☐LIC-ZZZ/English ☐Altro:................... | | | | | | | | | | | |

|  |  |
| --- | --- |
| **DOMANDA 1** | *Risultato* |
| Indicare il range di rappresentazione di numeri a 16 bit nella codifica in complemento a 2 e nella codifica in modulo e segno. Evidenziare le differenze. | Range complemento a due:  < (X)CA2 <  Range modulo e segno:  < (X)MS < |
| Principali differenze: | |

|  |
| --- |
| **DOMANDA 2** |
| Spiegare perché le due espressioni e non possono mai essere uguali |
| Risposta |

|  |
| --- |
| **DOMANDA 3** |
| Ordinare le principali tecnologie di memorie che si conoscono secondo la dimensione generalmente utilizzata nei sistemi di calcolo. |
| Risposta |

|  |
| --- |
| **DOMANDA 4 (Programmazione)** |

Scrivere un programma in C che esegua la codifica chiamata bus-invert. Tale codifica agisce su sequenze di parole di lunghezza N costituite da un alfabeto binario. Ogni parola è confrontata con la precedente e viene calcolata una distanza tra le due parole. La distanza si calcola aggiungendo 1 ogni volta che due cifre corrispondenti delle due parole sono diverse. Ad esempio (con parole di lunghezza 8):

11110000 00001111 distanza: 8

11110000 11110000 distanza: 0

11110000 11111111 distanza: 4

La codifica consiste nell’invertire tutti i simboli della parola se la distanza dalla precedente è strettamente maggiore di N/2 e lasciarli inalterati se la distanza è minore di N/2. Si noti che la distanza deve essere calcolata sempre rispetto alla codifica della parola precedente (quindi ad esempio se la parola precedente è stata invertita, la distanza deve essere calcolata rispetto alla parola invertita).

Il programma deve leggere da un file formato ASCII contenente per ogni riga una parola binaria di lunghezza N, con N costante definita tramite la direttiva #define. **Il file di input viene passato come primo argomento sulla linea di comando, ed ha lunghezza non nota a priori**. Il programma deve scrivere su un **file di output specificato come secondo argomento sulla linea di comando** le parole codificate e, separato da uno spazio, un carattere (I o N) che indichi se la parola è stata invertita oppure no. Si assuma che la prima parola non venga mai invertita.

Inoltre il programma **deve stampare su console la sequenza delle 3 parole consecutive avente la somma delle distanze complessive calcolate minima.** In caso esista più di una tripletta minima è sufficiente stamparne solo una.Si assuma che il formato del file di input sia corretto.

Esempio di file input e output (con N=8):

Nell’esempio sono indicate in grassetto, a titolo di esempio, i simboli diversi dalla parola precedente. Notare che il calcolo della distanza viene fatto sempre rispetto alla parola appena codificata. Le distanze calcolate nell’esempio sono rispettivamente: 6, 3, 2.

|  |  |
| --- | --- |
| parole.txt (input) | codifica.txt  (output) |
| 11110000  **00**11**1111**  11**111**000  11111**11**0 | 11110000 N  11000000 I  11111000 N  11111110 N |

Esempio di esecuzione:

C:\> **businvert parole.txt codifica.txt**

Sequenza di 3 parole consecutive con minima somma delle distanze calcolate: 00111111 11111000 11111110

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CORSO** | **Informatica – 12BHD** | | | | **DATA** | | **01/09/2014** | | | | |
| **NOME** |  | | | | **COGNOME** | |  | | | | |
| **MATRICOLA** | | **S** |  |  | |  | |  |  |  | **C/2** |
| ☐AAA-BARC ☐BARD-BOUH ☐BOUI-CART ☐CARU-CONS ☐CARU-CONS ☐CONT-DEMAR ☐DEMAS-FERRD ☐FERRE-GIAQ ☐GIAR-LAEZ ☐LAFA-MANC ☐MAND-MIQZ ☐MIRA-PAHZ  ☐PAIA-PODD ☐PODE-ROSSE ☐ROSSF-SIQZ ☐SIRA-TUCB ☐TUCC-ZZZ  ☐Poli@Home ☐5 Crediti ☐AAA-LIB/English ☐LIC-ZZZ/English ☐Altro:................... | | | | | | | | | | | |

|  |  |
| --- | --- |
| **DOMANDA 1** | *Risultato* |
| Indicare il range di rappresentazione di numeri a 32 bit nella codifica in complemento a 2 e nella codifica in modulo e segno. Evidenziare le differenze. | Range complemento a due:  < (X)CA2 <  Range modulo e segno:  < (X)MS < |
| Principali differenze: | |

|  |
| --- |
| **DOMANDA 2** |
| Spiegare perché le due espressioni e sono sempre uguali |
| Risposta |

|  |
| --- |
| **DOMANDA 3** |
| Ordinare le principali tecnologie di memorie che si conoscono secondo la velocità di accesso. |
| Risposta |

|  |
| --- |
| **DOMANDA 4 (Programmazione)** |

Scrivere un programma in C che esegua la codifica chiamata bus-invert. Tale codifica agisce su sequenze di parole di lunghezza N costituite da un alfabeto binario. Ogni parola è confrontata con la precedente e viene calcolata una distanza tra le due parole. La distanza si calcola aggiungendo 1 ogni volta che due cifre corrispondenti delle due parole sono diverse. Ad esempio (con parole di lunghezza 8):

11110000 00001111 distanza: 8

11110000 11110000 distanza: 0

11110000 11111111 distanza: 4

La codifica consiste nell’invertire tutti i simboli della parola se la distanza dalla precedente è strettamente maggiore di N/2 e lasciarli inalterati se la distanza è minore o uguale a N/2. Si noti che la distanza deve essere calcolata sempre rispetto alla codifica della parola precedente (quindi ad esempio se la parola precedente è stata invertita, la distanza deve essere calcolata rispetto alla parola invertita).

Il programma deve leggere da un file formato ASCII contenente per ogni riga una parola binaria di lunghezza N, con N costante definita tramite la direttiva #define. **Il file di input viene passato come primo argomento sulla linea di comando, ed ha lunghezza non nota a priori**. Il programma deve scrivere su un **file di output specificato come secondo argomento sulla linea di comando** le parole codificate e, separato da uno spazio, un carattere (I o N) che indichi se la parola è stata invertita oppure no. Si assuma che la prima parola non venga mai invertita.

Inoltre il programma **deve stampare su console la sequenza delle 3 parole consecutive avente la somma delle distanze complessive calcolate massima.** In caso esista più di una tripletta massima è sufficiente stamparne solo una. Si assuma che il formato del file di input sia corretto.

Esempio di file di input e output (con N=8):

Nell’esempio sono indicate in grassetto, a titolo di esempio, i simboli diversi dalla parola precedente. Notare che il calcolo della distanza viene fatto sempre rispetto alla parola appena codificata. Le distanze nell’esempio sono rispettivamente: 6, 3, 2.

|  |  |
| --- | --- |
| parole.txt (input) | codifica.txt  (output) |
| 11110000  **00**11**1111**  11**111**000  11111**11**0 | 11110000 N  11000000 I  11111000 N  11111110 N |

Esempio di esecuzione:

C:\> **businvert parole.txt codifica.txt**

Sequenza di 3 parole consecutive con massima somma delle distanze calcolate: 11110000 00111111 11111000