|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COGNOME:** |  | **NOME** |  | **D1** |
| **MATRICOLA:** |  | | |
| **DOCENTE:** |  | | |

|  |  |
| --- | --- |
| **Domanda 1** | Risultato |
| Si convertano i seguenti due numeri  da binario in complemento a 2 (ca2) e modulo e segno (MS) su 4 bit  a decimale:  1111ca2 = X10  0111MS = Y10 | X10 = \_\_\_\_\_\_\_\_\_\_\_\_  Y10 = \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Si riportino TUTTI i passaggi | |

|  |
| --- |
| **Domanda 2** |
| Si consideri una memoria di dimensione 200MB (megabyte) e address bus di 20 bit. Qual è il parallelismo della memoria in esame? |
|  |

|  |
| --- |
| **Domanda 3** |
| Spiegare le differenze tra memorie di tipo RAM e Hard Disk |
|  |

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

Un albergatore desidera gestire le prenotazioni del suo hotel tramite un programma. Tale programma deve espletare le seguenti funzioni:

* Ricevere come input il numero di camere e di ospiti per camera della corrente richiesta (questi parametri sono ricevuti da linea di comando come indicato successivamente)
* Verificare se esiste la disponibilità per tale richiesta leggendo la situazione corrente delle prenotazioni da file (il formato del file è specificato successivamente)
* Stampare a video le informazioni relative alla prenotazione, in particolare:
  + Se esiste disponibilità residua sufficiente, visualizzare messaggio di conferma e dettagli delle camere prenotate
  + In caso non sia possibile soddisfare la richiesta, visualizzare un messaggio di errore
* Aggiornare il file qualora la prenotazione possa essere effettuata.

Il primo parametro da linea di comando corrisponde al numero n di camere richiesta (>=1) mentre i successivi n parametri indicano la dimensione di ciascuna camera:

|  |
| --- |
| ESEMPIO parametri linea di comando:  **c:\> nomeprog 3 1 2 1**  Significa che si desidera prenotare 3 camere, da 1, 2 e 1 ospite ciascuna rispettivamente. |

Il file delle prenotazioni si chiama camere.txt ha il seguente formato:

* La prima linea indica in numero c di camere dell’albergo (nota bene: il programma può gestire alberghi fino ad un massimo di 100 stanze)
* Le successive c linee sono composte da
  + numero della camera
  + numero massimo di occupanti per la camera
  + la cifra ‘0’ se la camera è libera / ‘1’ se occupata

|  |
| --- |
| ESEMPIO codifica file camere.txt:  **5**  **1 2 0**  **2 3 1**  **3 2 0**  **4 1 0**  **5 2 1**  **6 1 0**  Significa che l’hotel ha 5 camere, la camera 1 può ospitare 2 persone ed è libera, la camera 2 può ospitare 3 persone ed è occupata, la camera 3 può ospitare 2 persone ed è libera, ecc... |

Il programma deve allocare ESATTAMENTE le richieste. Ad esempio, non è possibile assegnare a una richiesta singola (1 persona) una camera da 2 persone.

|  |  |
| --- | --- |
| ESEMPI di esecuzione programma basate sul contenuto di camere.txt riportato sopra: | |
| **c:\> nomeprog 3 1 2 1**  Prenotazione effettuata con successo  Camera 1 – 2 persone  Camera 4 – 1 persona  Camera 6 – 1 persona  Il file delle camere è stato aggiornato con successo | **c:\> nomeprog 3 2 2 3**  Prenotazione non effettuata – numero di camere libere insufficiente  Il file delle camere non è stato aggiornato |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COGNOME:** |  | **NOME** |  | **D2** |
| **MATRICOLA:** |  | | |
| **DOCENTE:** |  | | |

|  |  |
| --- | --- |
| **Domanda 1** | Risultato |
| Si convertano i seguenti due numeri  da binario in complemento a 2 (ca2) e modulo e segno (MS) su 4 bit  a decimale:  1111MS = X10  0111ca2 = Y10 | X10 = \_\_\_\_\_\_\_\_\_\_\_\_  Y10 = \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Si riportino TUTTI i passaggi | |

|  |
| --- |
| **Domanda 2** |
| Si consideri una memoria di dimensione 100MB (megabyte) e address bus di 10 bit. Qual è il parallelismo della memoria in esame? |
|  |

|  |
| --- |
| **Domanda 3** |
| Si elenchino i tipi di memoria conosciuti, suddividendoli tra memorie di tipo volatile e non-volatile |
|  |

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

Un albergatore desidera gestire le prenotazioni del suo hotel tramite un programma. Tale programma deve espletare le seguenti funzioni:

* Ricevere come input il numero di camere e di ospiti per camera della corrente richiesta (questi parametri sono ricevuti da linea di comando come indicato successivamente)
* Verificare se esiste la disponibilità per tale richiesta leggendo la situazione corrente delle prenotazioni da file (il formato del file è specificato successivamente)
* Stampare a video le informazioni relative alla prenotazione, in particolare:
  + Se esiste disponibilità residua sufficiente, visualizzare messaggio di conferma e dettagli delle camere prenotate
  + In caso non sia possibile soddisfare la richiesta, visualizzare un messaggio di errore
* Aggiornare il file qualora la prenotazione possa essere effettuata.

Il primo parametro da linea di comando corrisponde al numero n di camere richiesta (>=1) mentre i successivi n parametri indicano la dimensione di ciascuna camera:

|  |
| --- |
| ESEMPIO parametri linea di comando:  **c:\> nomeprog 3 1 2 1**  Significa che si desidera prenotare 3 camere, da 1, 2 e 1 ospite ciascuna rispettivamente. |

Il file delle prenotazioni si chiama camere.txt ha il seguente formato:

* La prima linea indica in numero c di camere dell’albergo (nota bene: il programma può gestire alberghi fino ad un massimo di 100 stanze)
* Le successive c linee sono composte da
  + numero della camera
  + numero massimo di occupanti per la camera
  + la lettera ‘L’ se la camera è libera / ‘O’ se occupata

|  |
| --- |
| ESEMPIO codifica file camere.txt:  **5**  **1 2 L**  **2 3 O**  **3 2 L**  **4 1 L**  **5 2 O**  **6 1 L**  Significa che l’hotel ha 5 camere, la camera 1 può ospitare 2 persone ed è libera, la camera 2 può ospitare 3 persone ed è occupata, la camera 3 può ospitare 2 persone ed è libera, ecc... |

Il programma deve allocare le richieste anche NON ESATTAMENTE. Ad esempio, è possibile assegnare a una richiesta singola (1 persona) una camera da 2 persone.

|  |  |
| --- | --- |
| ESEMPI di esecuzione programma basate sul contenuto di camere.txt riportato sopra: | |
| **c:\> nomeprog 3 1 2 1**  Prenotazione effettuata con successo  Camera 1 – 2 persone  Camera 4 – 1 persona  Camera 6 – 1 persona  Il file delle camere è stato aggiornato con successo | **c:\> nomeprog 3 2 2 3**  Prenotazione non effettuata – numero di camere libere insufficiente  Il file delle camere non è stato aggiornato |