|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NOME** |  | | | | **COGNOME** | |  | | | | |
| **MATRICOLA** | | **S** |  |  | |  | |  |  |  | **C/1** |
| **☐A-BARA ☐BARB-BOTS ☐BOTT-CAR ☐CAS-CORD ☐CORE-DIF ☐DIG-FIOR ☐FIOS-GIORD ☐GIORE-LANE ☐LANF-MARA ☐MORB-MOH ☐MOI-PAK ☐PAL-POLH  ☐POLI-ROSA ☐ROSB-SIL ☐SIM-TR ☐TS-ZZZ**  **☐Poli@Home ☐5 Crediti ☐English/A-L ☐English/M-Z ☐Altro:................** | | | | | | | | | | | |

|  |  |
| --- | --- |
| **DOMANDA 1** | *Risultato* |
| Dimostrare che la rappresentazione in CA2 di uno stesso numero varia al variare del numero di bit utilizzati. A tal scopo si utilizzino i due esempi:  n1= 60, su 7 e 8 bit  n2= -16, su 6 e 7 bit | n1(CA2) su 7 bit=  n1(CA2) su 8 bit=  n2(CA2) su 6 bit=  n2(CA2) su 7 bit= |
| Passaggi | |

|  |  |
| --- | --- |
| **DOMANDA 2** |  |
| Un’architettura è caratterizzata da un ADDRESS-BUS di 8 bit e un CONTROL-BUS di 4 bit. Considerando una memoria con parallelismo 8bit, calcolare la massima quantità di memoria indirizzabile. | |
| Risposta | |

|  |  |
| --- | --- |
| **DOMANDA 3** |  |
| Spiegare cosa si intende per linguaggio strutturato e tipizzato. | |
| Risposta | |
| **DOMANDA 4 (PROGRAMMAZIONE)** | |

Si scriva un programma per la gestione delle precedenze ad un incrocio di tre strade denominate N, S, E. Il programma deve ricevere gli arrivi a successivi istanti di tempo tramite un file passato come primo argomento da riga di commando. Ogni riga di tale file rappresenta un certo istante di tempo in questo modo:

**<TARGA\_N> <TARGA\_S> <TARGA\_E>**

Dove **TARGA** è una stringa di 7 caratteri, **TARGA\_N** indica un arrivo dalla strada N, **TARGA\_S** indica un arrivo dalla strada S, **TARGA\_E** indica arrivo dalla strada E. Nel caso in cui non ci sia una macchina proveniente da una determinata strada **TARGA** sarà rappresentata dalla stringa ‘XXXXXXX’.

Il numero di righe del file di arrivi non è noto a priori. Non è quindi possibile caricare i valori di tutte le targhe in memoria e nemmeno scorrere il file più volte.

Per ogni istante di tempo considerato (ovvero per ogni riga del file in ingresso) il programma deve scrivere su un file di output specificato come secondo argomento sulla riga di commando , la macchina che può attraversare l’incrocio secondo la seguente regola: **ha precedenza la macchina della fila più lunga**. Nel caso di file di ugual lunghezza si scelga la prima esaminata. Terminata la lettura del file in input continuare a scrivere in ordine le targhe delle macchine che attraversano l’incrocio fino all’esaurimento delle stesse.

Una macchina che non passa ad un certo istante di tempo all’incrocio rimane in fila. I nuovi arrivi di macchine ad ogni istante di tempo si accodano alle rispettive file.

L’assenza di una o più macchine nelle righe del file garantisce che la lunghezza massima della coda non superi un valore N prefissato, ad esempio pari a 100, e definito tramite #define.

**Esempio file arrivi.txt**

AA100BB CC200FF GG300HH

CC100ZZ YY200PP VV500KK

NN200KK XXXXXXX HH200DD

XXXXXXX FF600PP XXXXXXX

**Esempio esecuzione:**

**#> incrocio.exe arrivi.txt precedenze.txt**

**File precedenze.txt**

AA100BB

CC200FF

GG300HH

CC100ZZ

YY200PP

VV500KK

NN200KK

FF600PP

HH200DD

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NOME** |  | | | | **COGNOME** | |  | | | | |
| **MATRICOLA** | | **S** |  |  | |  | |  |  |  | **C/2** |
| **☐A-BARA ☐BARB-BOTS ☐BOTT-CAR ☐CAS-CORD ☐CORE-DIF ☐DIG-FIOR ☐FIOS-GIORD ☐GIORE-LANE ☐LANF-MARA ☐MORB-MOH ☐MOI-PAK ☐PAL-POLH  ☐POLI-ROSA ☐ROSB-SIL ☐SIM-TR ☐TS-ZZZ**  **☐Poli@Home ☐5 Crediti ☐English/A-L ☐English/M-Z ☐Altro:................** | | | | | | | | | | | |

|  |  |
| --- | --- |
| **DOMANDA 1** | *Risultato* |
| Dimostrare che la rappresentazione in CA2 di uno stesso numero varia al variare del numero di bit utilizzati. A tal scopo si utilizzino i due esempi    n1= -24, su 7 e 8 bit  n2= 19, su 6 e 7 bit | n1(CA2) su 7 bit=  n1(CA2) su 8 bit=  n2(CA2) su 6 bit=  n2(CA2) su 7 bit= |
| Passaggi | |

|  |  |
| --- | --- |
| **DOMANDA 2** |  |
| Considerata un’architettura in cui tutti i bus (ADDRESS-, DATA- e CONTROL-BUS) sono su 8 bit, calcolare la massima quantità di memoria indirizzabile. | |
| Risposta | |

|  |  |
| --- | --- |
| **DOMANDA 3** |  |
| Descrivere la differenza tra un linguaggio ad alto livello ed un linguaggio assembler. Nello specifico dire quale dei due agevola la programmazione e perchè. | |
| Risposta | |
| **DOMANDA 4 (PROGRAMMAZIONE)** | |

Si scriva un programma per la gestione delle precedenze ad un incrocio di tre strade. Il programma deve ricevere gli arrivi sulle varie file tramite un file passato come primo argomento da riga di commando. Ogni riga di tale file ha il seguente formato:

**<TARGA1> <TEMPO1> <TARGA2> <TEMPO2> <TARGA3> <TEMPO3>**

Dove **TARGA** è una stringa di 7 caratteri, mentre **TEMPO** è una stringa che rappresenta un orario sulle 24 ore: **ORE:MINUTI**. Le macchine in arrivo si distribuiscono secondo il seguente schema: nella strada 1 si accodano le macchine relative a **TARGA1**, nella strada 2 le macchine relative a **TARGA2**, nella strada 3 le macchine relative a **TARGA3**. Non si possono fare assunzioni sull’ordine temporale tra una riga e l’altra. Tuttavia all’interno di una stessa fila gli arrivi sono in orari successivi.

Il numero di righe del file di arrivi non è noto a priori, non è quindi possibile caricare tutte le informazioni in memoria e nemmeno scorrere il file più volte.

Per ogni istante di tempo considerato (ovvero per ogni riga del file in ingresso) il programma deve scrivere su un file di output specificato come secondo argomento sulla riga di commando, la macchina che può attraversare l’incrocio secondo la seguente regola: **ha precedenza la macchina che è arrivata per prima tra quelle già in coda all’incrocio**. Si assuma il file corretto e si assuma che non ci possano essere casi di arrivi contemporanei (stesso orario). Terminata la lettura del file in input continuare a scrivere in ordine le targhe delle macchine che attraversano l’incrocio fino all’esaurimento delle stesse.

Una macchina che non passa ad un certo istante di tempo all’incrocio rimane in fila. I nuovi arrivi di macchine ad ogni istante di tempo si accodano alle rispettive file.

Si garantisce che la lunghezza massima della coda non superi un valore N prefissato, ad esempio pari a 100, e definito tramite #define.

**Esempio file arrivi.txt**

AA100BB 11:23 CC200FF 11:28 GG300HH 11:29

CC100ZZ 11:24 YY200PP 11:35 VV500KK 11:30

NN200KK 11:34 BB300PP 12:00 HH200DD 12:23

**Esempio esecuzione:**

**#>incrocio.exe arrivi.txt precedenze.txt**

**File precedenze.txt**

AA100BB

CC100ZZ

CC200FF

GG300HH

VV500KK

NN200KK

YY200PP

BB300PP

HH200DD