Soluzione Esercizio 2

Scopo dell'esercizio è implementare una funzione *push()* che sia in grado di inserire un valore intero *key* all'interno di una pila scelta tra una di quelle presenti in un vettore di pile. Per poter individuare la pila da utilizzare e quindi la cella del vettore contenente il puntatore alla testa della pila si utilizza una funzione chiamata *posizione()* la quale prende in ingresso il valore *key* da inserire in una pila e restituisce un intero che individua la pila da utilizzare o meglio l'indice della cella dell'array contenente la testa della pila che deve essere utilizzata.

Quindi se consideriamo un vettore di 100 celle (e quindi con 100 pile) se dobbiamo inserire il valore intero pari a key = 23432, chiameremo la funzione *posizione()* passando il valore 23432 per sapere in quale delle 100 pile dobbiamo eseguire l'inserimento. Come detto sopra il valore restituito dalla funzione *posizione()* sarà un intero, in questo esempio un valore tra 0 e 99, ad indicarci la cella da utilizzare.

Vediamo la soluzione:

Versione base (ma valida per avere il massimo del punteggio nell'esercizio!):

```
struct cella
{
    int valore;
    struct cella *next;
};

struct cella *elencoPile[37];


int posizione(int key)
{
    return key % 37;
}


void push(int key)
{
    int pos;
    struct cella *nuovo;

    nuovo = (struct cella*) malloc(sizeof(struct cella));
    nuovo->valore = key;

    pos = posizione(key);  //individuo la pila da dover utilizzare

    //inserisco l'elemento in testa alla pila
    nuovo->next = elencoPile[pos];
    elencoPile[pos] = nuovo;
}
```


Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it
http://www.programmazione.info

Versione più completa:

```
#define MAX 37

struct cella
{
    int valore;
    struct cella *next;
};

struct cella *elencoPile[MAX];


int posizione(int key)
{
    return key % MAX;   // MAX = 37
}


void push(int key)
{
    int pos;
    struct cella *nuovo;

    nuovo = (struct cella*) malloc(sizeof(struct cella));

    if (nuovo == NULL)
    {
        printf("Errore nell\'allocazione della memoria");
        exit(1);
    }

    nuovo->valore = key;

    pos = posizione(key);  //individuo la pila da dover utilizzare

    if (pos < MAX)
    {
        //inserisco l'elemento in testa alla pila
        nuovo->next = elencoPile[pos];
        elencoPile[pos] = nuovo;
    }
    else
    {
        printf("Indice non valido");
        exit(1);
    }
}
```


Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it
http://www.programmazione.info

Soluzione Esercizio 3

Nella soluzione proposta considereremo la matricola 164795:

la matrice "alfabeto" e il vettore "array" vengono passati alle varie chiamate ricorsive senza essere modificati quindi non li terremo in considerazione in ogni passaggio durante la spiegazione:

```
1     char cruciverba(char mat[MAX_M][MAX_M], int vet[MAX_AR], int k, int i, int
2     j)
3     {
4        if (k >= MAX_AR-1)
5           return mat[i][j];
6
7        i = vet[k++] % MAX_M;
8        j = vet[k] % MAX_M;
9
10       if ((vet[k] % 2) == 0)
11          printf("%c ", cruciverba(mat,vet,k,i,j));
12       else
13          printf("%c ", cruciverba(mat,vet,k,j,i));
14
15       return mat[j][i];
      }
```

la prima chiamata presente nel main causa l'interruzione del programma e il passaggio del controllo alla prima istanza della funzione cruciverba:

Istanza 1
entrati nella prima istanza:

k = 0
i = 0
j = 0

considerando la riga 3: dato che MAX_AR = 6 e che MAX_AR – 1 = 5 la variabile k essendo uguale a 0 è minore di 5 quindi si salta il corpo dell'istruzione if

si procede eseguendo la riga 6:

i = vet[k++] % MAX_M;

equivale a:

i = vet[0] % 5;

NB. la variabile "k" non viene incrementata subito in quanto l'operatore unario ++ si trova dopo la variabile, quindi prima si considera la variabile con il suo valore poi si esegue l'incremento

considerando la matricola 164795 nella cella 0 del vettore troviamo il numero 1 quindi l'espressione diventa:

i = 1 % 5

alla fine nella variabile "i" avremo come valore 1 (resto della divisione di: 1 diviso 5)

si procede allo stesso modo con la variabile "j", quindi con la riga 7:

Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it
http://www.programmazione.info

```
j = vet[k] % MAX_M;
```

equivale a:

```
j = vet[1] % 5;
```

NB. la variabile "k" viene modificata dopo essere utilizzata nel calcolare "i" quindi avremo k = 1.

considerando la matricola 164795 nella cella 1 del vettore troviamo il numero 6 quindi l'espressione diventa:

```
j = 6 % 5
```

alla fine nella variabile "j" avremo come valore 1 (resto della divisione di: 6 diviso 5)

con l'istruzione if presente nella riga 9 valutiamo il valore della cella del vettore con indice k (k = 1), considerando la matricola 164795 nella cella 1 del vettore troviamo il numero 6, procediamo con l'eseguire il corpo dell'istruzione if (riga 10) dato che 6 è numero pari.

L'istanza 1 della funzione cruciverba si blocca e cede il controllo alla seconda chiamata ricorsiva passando i valori
k = 1, i = 1, j = 1

Istanza 2

entrati nella seconda istanza, considerando i valori passati durante la chiamata avremo:

```
k = 1
i = 1
j = 1
```

considerando la riga 3: dato che MAX_AR = 6 e che MAX_AR – 1 = 5 la variabile k essendo uguale a 1 è minore di 5 quindi si salta il corpo dell'istruzione if

si procede eseguendo la riga 6:

```
i = vet[k++] % MAX_M;
```

equivale a:

```
i = vet[1] % 5;
```

NB. la variabile "k" non viene incrementata subito in quanto l'operatore unario ++ si trova dopo la variabile, quindi prima si considera la variabile con il suo valore poi si esegue l'incremento

considerando la matricola 164795 nella cella 1 del vettore troviamo il numero 6 quindi l'espressione diventa:

```
i = 6 % 5
```

alla fine nella variabile "i" avremo come valore 1 (resto della divisione di: 6 diviso 5)

si procede allo stesso modo con la variabile "j", quindi con la riga 7:

```
j = vet[k] % MAX_M;
```


Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it
http://www.programmazione.info

equivale a:

```
j = vet[2] % 5;
```

NB. la variabile "k" viene modificata dopo essere utilizzata nel calcolare "i" quindi avremo il k = 2. considerando la matricola 164795 nella cella 2 del vettore troviamo il numero 4 quindi l'espressione diventa:

```
j = 4 % 5
```

alla fine nella variabile "j" avremo come valore 4 (resto della divisione di: 4 diviso 5)

con l'istruzione if presente nella riga 9 valutiamo il valore della cella del vettore con indice k (k = 2), considerando la matricola 164795 nella cella 2 del vettore troviamo il numero 4, procediamo con l'eseguire il corpo dell'istruzione if (riga 10) dato che 4 è numero pari.

L'istanza 2 della funzione cruciverba si blocca e cede il controllo alla terza chiamata ricorsiva passando i valori
k = 2, i = 1, j = 4

Istanza 3

entrati nella terza istanza, considerando i valori passati durante la chiamata avremo:

```
k = 2
i = 1
j = 4
```

considerando la riga 3: dato che MAX_AR = 6 e che MAX_AR – 1 = 5 la variabile k essendo uguale a 2 è minore di 5 quindi si salta il corpo dell'istruzione if

si procede eseguendo la riga 6:

```
i = vet[k++] % MAX_M;
```

equivale a:

```
i = vet[2] % 5;
```

NB. la variabile "k" non viene incrementata subito in quanto l'operatore unario ++ si trova dopo la variabile, quindi prima si considera la variabile con il suo valore poi si esegue l'incremento

considerando la matricola 164795 nella cella 2 del vettore troviamo il numero 4 quindi l'espressione diventa:

```
i = 4 % 5
```

alla fine nella variabile "i" avremo come valore 4 (resto della divisione di: 4 diviso 5)

si procede allo stesso modo con la variabile "j", quindi con la riga 7:

```
j = vet[k] % MAX_M;
```

equivale a:

```
j = vet[3] % 5;
```


Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it
http://www.programmazione.info

NB. la variabile "k" viene modificata dopo essere utilizzata nel calcolare "i" quindi avremo k = 3.

considerando la matricola 164795 nella cella 3 del vettore troviamo il numero 7 quindi l'espressione diventa:

```
j = 7 % 5
```

alla fine nella variabile "j" avremo come valore 2 (resto della divisione di: 7 diviso 5)
con l'istruzione if presente nella riga 9 valutiamo il valore della cella del vettore con indice k (k = 3), considerando la matricola 164795 nella cella 3 del vettore troviamo il numero 7, procediamo con l'eseguire il corpo dell'istruzione else (riga 12) dato che 7 è numero dispari.

L'istanza 3 della funzione cruciverba si blocca e cede il controllo alla quarta chiamata ricorsiva passando i valori
k = 3, i = 4, j = 2

NB. FACCIAMO ATTENZIONE IN QUANTO LE VARIABILI "i" e "j" vengono passate invertite, quindi nella quarta istanza "i" prende il valore di "j" (terza istanza) e "j" prende il valore di "i" (terza istanza)

Istanza 4
entrati nella quarta istanza, considerando i valori passati durante la chiamata avremo:
NB. FACCIAMO ATTENZIONE IN QUANTO LE VARIABILI "i" e "j" vengono passate invertite

```
k = 3
i = 2
j = 4
```

k < 5 quindi si procede eseguendo la riga 6:

```
i = vet[k++] % MAX_M;
```

equivale a:

```
i = vet[3] % 5;
```

alla fine nella variabile "i" avremo come valore 2 (resto della divisione di: 7 diviso 5)

si procede allo stesso modo con la variabile "j", quindi con la riga 7:

```
j = vet[k] % MAX_M;
```

equivale a:

```
j = vet[4] % 5;
```

alla fine nella variabile "j" avremo come valore 4 (resto della divisione di: 9 diviso 5)

vet[4] è dispari quindi si passa alla riga 12 passando alla chiamata ricorsiva i valori: k = 4, i = 2, j = 4
NB. FACCIAMO ATTENZIONE IN QUANTO LE VARIABILI "i" e "j" vengono passate invertite, quindi nella quinta istanza "i" prende il valore di "j" (quarta istanza) e "j" prende il valore di "i" (quarta istanza)

Istanza 5

entrati nella quinta istanza, considerando i valori passati durante la chiamata avremo:
NB. FACCIAMO ATTENZIONE IN QUANTO LE VARIABILI "i" e "j" vengono passate invertite

k = 4
i = 4
j = 2

k < 5 quindi si procede eseguendo la riga 6:

```
i = vet[k++] % MAX_M;
```

equivale a:

```
i = vet[4] % 5;
```

alla fine nella variabile "i" avremo come valore 4 (resto della divisione di: 9 diviso 5)

si procede allo stesso modo con la variabile "j", quindi con la riga 7:

```
j = vet[k] % MAX_M;
```

equivale a:

```
j = vet[5] % 5;
```

alla fine nella variabile "j" avremo come valore 0 (resto della divisione di: 5 diviso 5)

vet[5] è dispari quindi si passa alla riga 12 passando alla chiamata ricorsiva i valori: k = 5, i = 4, j = 0

Istanza 6

entrati nella sesta istanza, considerando i valori passati durante la chiamata avremo:
NB. FACCIAMO ATTENZIONE IN QUANTO LE VARIABILI "i" e "j" vengono passate invertite

k = 5
i = 0
j = 4

k = 5 quindi si procede eseguendo la riga 4, restituendo al chiamante la lettera presente all'interno della matrice: mat[i][j], considerando i valori passati e cioè i = 0 e j = 4 avremo mat[0][4] quindi la lettera **F**
il valore viene restituito all'istanza 5

Risaliamo nelle ricorsione:

- La quinta istanza riceve il valore **F** dalla sesta e lo stampa a video, si procede poi con l'eseguire la riga 14 (facciamo attenzione al fatto che siamo all'interno della quarta istanza quindi dobbiamo considerare lo stato delle variabili in tale chiamata lasciata in sospeso, quindi i = 4 e j = 0; i valori di "i" e "j" devono essere considerati quelli ottenuti dai calcoli nelle righe 6 e 7).
  Nella riga 14 viene restituita la lettera presente all'interno di mat[j][i] (notare "i" e "j" invertiti), quindi mat[0][4] quindi restituiremo il valore **F** alla quarta istanza

- La quarta istanza riceve il valore **F** dalla quinta e lo stampa a video, si procede poi con l'eseguire la riga 14 (facciamo attenzione al fatto che siamo all'interno della quarta istanza quindi dobbiamo considerare lo stato delle variabili in tale chiamata lasciata in sospeso, quindi i = 2 e j = 4; i valori di "i" e "j" devono essere considerati quelli ottenuti dai calcoli nelle righe 6 e 7).
  Nella riga 14 viene restituita la lettera presente all'interno di mat[j][i] (notare "i" e "j" invertiti), quindi mat[4][2] quindi restituiremo il valore **T** alla terza istanza

- La terza istanza riceve il valore **T** dalla quarta e lo stampa a video, si procede poi con l'eseguire la riga 14 (facciamo attenzione al fatto che siamo all'interno della terza istanza quindi dobbiamo considerare lo stato delle variabili in tale chiamata lasciata in sospeso, quindi i = 4 e j = 2; i valori di "i" e "j" devono essere considerati quelli ottenuti dai calcoli nelle righe 6 e 7).
  Nella riga 14 viene restituita la lettera presente all'interno di mat[j][i] (notare "i" e "j" invertiti), quindi mat[2][4] quindi restituiremo il valore **Z** alla seconda istanza

- La seconda istanza riceve il valore **Z** dalla terza e lo stampa a video, si procede poi con l'eseguire la riga 14 (facciamo attenzione al fatto che siamo all'interno della seconda istanza quindi dobbiamo considerare lo stato delle variabili in tale chiamata lasciata in sospeso, quindi i = 1 e j = 4; i valori di "i" e "j" devono essere considerati quelli ottenuti dai calcoli nelle righe 6 e 7).
  Nella riga 14 viene restituita la lettera presente all'interno di mat[j][i] (notare "i" e "j" invertiti), quindi mat[4][1] quindi restituiremo il valore **K** alla seconda prima

- La prima istanza riceve il valore **K** dalla seconda e lo stampa a video, si procede poi con l'eseguire la riga 14 (facciamo attenzione al fatto che siamo all'interno della prima istanza quindi dobbiamo considerare lo stato delle variabili in tale chiamata lasciata in sospeso, quindi i = 1 e j = 1; i valori di "i" e "j" devono essere considerati quelli ottenuti dai calcoli nelle righe 6 e 7).
  Nella riga 14 viene restituita la lettera presente all'interno di mat[j][i] (notare "i" e "j" invertiti), quindi mat[1][1] quindi restituiremo il valore **Q** al main il quale provvederà a stampare il valore.

QUINDI RIASSUMENDO AVREMO LA STAMPA DELLE LETTERE:

F F T Z K Q

per chiarimenti e commenti in merito alle soluzioni proposte contattare:

Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it