

IV Appello 11-06-2014 Soluzione

Esercizio 1

```
void eliminaMinimo(elem*& testa) {
    if (testa== NULL) return ;
    int min = testa->info ;
    elem*q ;
    for(q=testa->pun; q !=NULL ; q=q->pun)
        if(q->info < min)
            min = q->info ;

    q=testa ;
    while((q !=NULL) && (q->info == min)) {
        testa=testa->pun ;
        delte q ;
        q=testa ;
    }

    if (q==NULL) return;

    elem* prec = q ;
    q = q->pun ;
    while (q !=NULL)
        if (q->info==min) {
            prec->pun = q->pun ;
            delete q ;
            q=prec->pun ;
        }
        else {
            p=q ;
            q= q->pun ;
        }
    return ;
}
```

Esercizio 3

```
int contaMaiuscole(char* v, int n) {
    if (n==0)
        return 0;
    if ((v[n-1] >= 'A') && (v[n-1] <= 'z')) {
        v[n-1] = '?';
        return 1 + contaMaiuscole(v, n-1);
    }
    return contaMaiuscole(v, n-1);
}
```

Esercizio 2

```
int* creatrice(const char* s) {

    ifstream in(s);
    int dim;
    int* mat;
    in>>dim;
    if((!in) || (dim <=0)){
        mat = new int [9];
        for(int i=0; i<3; i++)
            for(int j=0; j<3; j++){
                if((j==i) || (j==2-i))
                    mat[i*3+j] = 1;
                else
                    mat[i*3+j] = 0;
            }
        return mat;
    }

    mat = new int[dim*dim];
    int numero;
    for(int i=0; i<dim; i++)
        for(int j=0; j<dim; j++){
            if(in >> numero)
                mat[i*dim+j] = numero;
            else
                mat[i*dim+j] = 0;
        }
    return mat;
}
```

Esercizio 4

4.1

- a) Si: FFFD corrisponde a -3, FFFE a -2
- b) Si
- c) Si $(15 \cdot 18^3 + 15 \cdot 18^2 + 15 \cdot 18 + 13/14)$

4.2

Il programma dopo aver messo in AL gli otto bit corrispondenti alle 2 cifre meno significative del proprio numero di matricola inizializza CL a 8 e BL a zero. Successivamente, all'interno di un ciclo ripetuto 8 volte, viene shiftato a sinistra BL e a destra AL. Se il precedente bit meno significativo di AL non conteneva un 1, dopo lo shift non viene settato il flag di carry e pertanto si salta a LAB, dove si ritorna a ripetere il ciclo (BL ha subito solo una traslazione a sinistra ed e' entrato uno zero da destra). Viceversa se il precedente bit meno significativo di AL era 1, il bit di carry viene settato e dunque non si salta a LAB, bensì si esegue l'istruzione successiva che mette un 1 nel bit meno significativo di AL. Dopo otto ripetizioni nel registro BL si troverà una copia speculare del contenuto di AL (come se il contenuto fosse stato guardato allo specchio).

Pertanto ad esempio nel caso di ingresso "34" verrà prodotto

in uscita "2C" (0011-0100 => 0010-1100, ossia "2C", per l'appunto).

Una volta calcolato lo speculare di ciascuna cifra da 0 a 9, l'uscita del programma sono le rispettive cifre speculari, in ordine speculare:

0	=	0000	=>	0000	=	0
1	=	0001	=>	1000	=	8
2	=	0010	=>	0100	=	4
3	=	0011	=>	1100	=	C
4	=	0100	=>	0010	=	2
5	=	0101	=>	1010	=	A
6	=	0110	=>	0110	=	6
7	=	0111	=>	1110	=	E
8	=	1000	=>	0001	=	1
9	=	1001	=>	1001	=	9

Esempi:

12 => 48

35 => AC