

- 1)
- ```
int quanti(elem* p0){
 if(p0==NULL) return 0;
 if((p0->info >= 'a') && (p0->info <= 'z')) return 1+quanti(p0->pun);
 return quanti(p0->pun);
}
```
- 2)
- ```
void ins(elem*& testa, int n){
    if(n<=0) return;
    elem* p = testa;
    //appendo il primo elemento
    if(testa==NULL){ //la lista circolare era vuota
        testa = new elem;
        testa->info = 100;
        p = testa;
    }
    else{
        elem* q = p->pun;
        //raggiungo la coda della lista circolare
        while(q!=testa){
            p = q;
            q = q->pun;
        }
        //p e' sull'ultimo elem della lista circolare
        p->pun = new elem;
        p = p->pun;
        p->info = 100;
    }
    //creo il resto degli elementi
    for(int i=1; i<n; i++){
        p->pun = new elem;
        p = p->pun;
        p->info = 100;
    }
    //chiusura lista circolare
    p->pun = testa;
}
```
- 3)
- ```
int* unisci(int* v1, int dim1, int* v2, int dim2){
 int* vett = new int[dim1+dim2];
 int pos1 = dim1-1; //indice dell'ultimo elem del vettore v1
 int pos2 = dim2-1; //indice dell'ultimo elem del vettore v2
 for(int i=0; i<(dim1+dim2); i++){
 if((pos2<0) || ((pos1>=0) && (v1[pos1]>v2[pos2]]))){
 vett[i] = v1[pos1];
 pos1--;//diventa -1 quando ho esaurito gli elem
 }
 else{
 vett[i] = v2[pos2];
 pos2--;//diventa -1 quando ho esaurito gli elem
 }
 }
 return vett;
}
```

4)

(A09B)base16 = (1010 0000 1001 1011)base2  
(-100)base10 = (10011100)compl2

5)

alpha:a=10  
beta:a=20  
alpha:a=10  
beta:a=1  
gamma:a=20  
alpha:stampa() a=21  
gamma:stampa() a=21  
gamma:stampa() a=22  
gamma :distr  
beta:distr  
alpha:distr  
beta:distr  
alpha:distr

6)

input: 10 10  
nuova eccezione  
ecc1  
ecc main: 1

input: 20 5  
nuova eccezione  
ecc2  
ecc f: 2  
0

input: 20 10  
ecc generica