

Soluzione delle due parti di programmazione dell'esam del 12/7

```
struct triple{int L,TT, PP; triple(int a=0, int b=0, int c=0){L=a;TT=b;PP=c;}};
```

```
//Parte iterativa
```

```
int match1(int*T, int dimT, int*P, int dimP)
{
    bool fail=false;
    int i=0;
    while( i<dimT && i<dimP &&!fail)
    {
        if(P[i]!=T[i])
            fail=true;
        else
            i++;
    }
    return i;
}
//Post=(restituisce la lunghezza del massimo match che inizia in T e in P)
```

```
triple match(int*T, int dimT, int*P, int dimP)
{
    int bestP=-1, bestL=0, bestT=-1;
    for(int i=0; i<dimT-bestL; i++)
    {
        for(int j=0; j<dimP; j++)
        {
            int x=match1(T+i,dimT-i,P+j,dimP-j);
            if(x > bestL)
                {bestP=j;bestL=x; bestT=i;}
        }
    }
    return triple(bestL,bestT,bestP);
}
```

```
//Parte Ricorsiva
```

```
triple match1_ric(int*T, int dimT, int iT,int*P, int dimP,int iP);
int match0_ric(int*T, int dimT, int *P, int dimP);
```

```
triple matchR(int*T,int dimT, int iT, int*P, int dimP, int iP)
{
    if(iT==dimT) return triple();
    triple x= matchR(T,dimT,iT+1,P,dimP,iP);

    triple y= match1_ric(T,dimT,iT,P,dimP,iP);
    if(x.L> y.L)
        return x;
    else
        return y;
}
```

```

triple match1_ric(int*T, int dimT, int iT,int*P, int dimP,int iP)
{
    if(iP==dimP) return triple();
    triple x= match1_ric(T,dimT, iT,P,dimP,iP+1);
    int y= match0_ric(T+iT,dimT-iT,P+iP,dimP-iP);
    if(x.L > y)
        return x;
    else
        return triple(y,iT,iP);
}
//POST=(restituisce la trpla che corrisponde al migliore match che inizia in T[iP] e considera tutti i suffissi
// di P a partire da P[Ip])

int match0_ric(int*T, int dimT, int *P, int dimP)
{
    if(!dimT || !dimP)
        return 0;
    if(*T== *P)
        return 1 + match0_ric(T+1,dimT-1,P+1,dimP-1);
    return 0;
}
//post=(restituisce la lunghezza del miglior match che parte da T[Ip] e da P[Ip])

```