

Late For Work (time)	Graduation Card (text)
<pre> #include <iostream> using namespace std; #include <fstream> int main() { ifstream fr("input.txt"); ofstream fw("output.txt"); short int H0, M0, H1, M1, M0w, M1w; fr >> H0 >> M0 >> H1 >> M1; M0w = H0*60 + M0; M1w = H1*60 + M1; if ((H0 > H1) ((H0 == H1) && M0 > M1)) M1w = 24*60 + M1w; M1w = M1w - M0w; fw << M1w/60 << ' ' << M1w%60; } </pre>	<pre> #include <iostream> using namespace std; #include <fstream> #include <fstream> #define SP ' ' string W ; int main(){ ifstream fr("input.txt"); ofstream fw("output.txt"); int N, K, Wlen, Rlen; fr >> N >> K; Rlen=0; for (int i=0; i <= N-1; i++){ fr >> W; Wlen = W.length(); if (Rlen > 0) if ((Rlen + Wlen) <= K){ fw << W << SP; Rlen = Rlen + Wlen +1; } else { fw << endl << W << SP; Rlen = Wlen +1; } else { fw << W << SP; Rlen = Wlen +1; } } } </pre>

Science against spam (spam) – 1^parte	Science against spam (spam) – 2^parte
<pre> #include <iostream> using namespace std; #include <fstream> #define SP ' ' string BW [1000]; string GW [1000]; int main(){ ifstream fr("input.txt"); ofstream fw("output.txt"); int E, ENW, nESpam=0, nELeg=0; int B, G, j, k; string W; char TB = 'N', TG = 'N'; fr >> B; for (int i=0; i<B; i++) fr >> BW [i]; fr >> G; for (int i=0; i<G; i++) fr >> GW [i]; fr >> E; </pre>	<pre> for (int i=0; i < E; i++){ fr >> ENW; // Email Number Words TB = 'N'; TG = 'N'; for (j=0; j < ENW; j++){ fr >> W; k=0; while (k<B && TB == 'N'){ if (W == BW[k]){ TB = 'S'; } k++; } k=0; while (k<G && TG == 'N'){ if (W == GW[k]){ TG = 'S'; } k++; } } //----- ENW----- if (TB == 'S' && TG == 'N'){ nESpam++; } else if (TB == 'N' && TG == 'S') nELeg++; } fw << nESpam << SP << nELeg; } </pre>

Discesa massima (discesa)	Numeri di Figonacci (figonacci)
<pre> #include <iostream> using namespace std; #include <fstream> #define SP '\t' #define N 10 int main() { ifstream fr("input.txt"); ofstream fw("output.txt"); int A, i, j, MAX, temp; int Pir[N][N], Psum[N][N]={0}; fr >> A; for (i=0; i < A; i++) for (j=0; j <= i; j++) fr >> Pir[i][j]; Psum[0][0] = Pir[0][0]; for (i=1; i < A; i++){ for (j=0; j < i; j++){ temp = Pir[i][j] + Psum[i-1][j]; if (Psum[i][j] < temp) Psum[i][j] = temp; temp = Pir[i][j+1] + Psum[i-1][j]; if (Psum[i][j+1] < temp) Psum[i][j+1] = temp; } MAX = Psum[A-1][0]; for (j=1; j < N; j++) if (Psum[A-1][j] > MAX) MAX = Psum[A-1][j]; fw << MAX; } </pre>	<pre> #include <iostream> #include <fstream> using namespace std; int main() { ifstream in ("input.txt"); ofstream out("output.txt"); int N, M, G, G_1, som, som_2, som_1; in >> N >> M; G_1 = 1; G = 1; // G[2] som_2 = - 1; // G[0] + G[1] som_1 = 0; // G[0] + G[1] + G[2] for (int i=3; i <= N; i++){ G = ((i-1)%M * G_1%M - som_2%M + M)%M; som = (som_1%M + G%M + M)%M; som_2 = som_1; som_1 = som; G_1 = G; } out << G; return 0; } </pre>

Crittografia (crittografia) – 1^parte	Crittografia (crittografia) – 2^parte
<pre> #include <iostream> #include <fstream> using namespace std; #define MAX 200000 char Sinp[MAX/2+1]; char password[11]={' ', nextC, old}; short int POS[10]={0}, N, indP=0; #define IMP "impossibile" short int cntlPOS(char x, char y, int k){ short int cifra = x-48; if (y >=48 && y <= 57) //-- anche y numerico return 1; if (POS[cifra] != 0 && password[cifra] != y){ POS[cifra]++; return 1; } password[cifra] = y; POS[cifra]++; return 0; } main(){ password[10]='\0'; int L, i, med, simm; ifstream fr("input.txt"); ofstream fw("output.txt"); fr >> N >> L; med=L/2; Sinp[med]='\0'; for (i=0; i<med; i++) fr >> Sinp[i]; </pre>	<pre> for (i=med; i<L; i++){ fr >> nextC; simm=L-1-i; old=Sinp[simm]; if (nextC != old){ if (old >= 48 && old <= 57){ //-- tra 0 e 9 if (cntlPOS(old, nextC, 2) == 1){ fw << IMP; return 0; } } else if (nextC >= 48 && nextC <= 57){/-- tra 0 e 9 if (cntlPOS(nextC, old, 1) == 1){ fw << IMP; return 0; } } else { fw << IMP << 3<<i; return 3; } } else if (nextC >= 48 && nextC <= 57){ /---2 numeri fw << IMP; return 0; } } } fw << password; } </pre>