1. Ανοίξτε ένα τερματικό στον Η/Υ σας.

2. Χρησιμοποιώντας την εντολή **mkdir**, κατασκευάστε κάτω από το home directory σας (~) την παρακάτω δομή **φακέλων**:

```
~/test1/
├── data
│   ├── binary
│   └── text
├── include
│   ├── m_files
│   ├── other_files
│   └── s_files
├── log
└── src
```

Με την εντολή **tree** τυπώστε στην οθόνη τη δομή που δημιουργήσατε.

Σε περίπτωση που δεν είναι αυτή του παραπάνω διαγράμματος (ενδιαφέρει η δομή και όχι η σειρά εκτύπωσης που συνήθως είναι αλφαβητική), θα πρέπει να χρησιμοποιήσετε τις εντολές **mkdir**, **mv**, **rm** και/ή **rmdir** για να κάνετε τις απαραίτητες διορθώσεις.

3. Ανακατεύθυνση της εξόδου μίας εντολής σε ένα αρχείο: **εντολή > αρχείο**

π.χ. `tree ~ > /tmp/myfile.txt`

Χρησιμοποιώντας την ανακατεύθυνση εξόδου, αποθηκεύστε την έξοδο της εντολής **tree** για τον φάκελο **~/test1** στο αρχείο **out3.log** μέσα στον φάκελο **~/test1/log**

Η εντολή **history** εμφανίζει τις εντολές που έχετε δώσει έως τώρα. Χρησιμοποιώντας την ανακατεύθυνση εξόδου, αποθηκεύστε την έξοδο της εντολής history στο αρχείο **his3.log** μέσα στον φάκελο **~/test1/log**

4. Ανακατεύθυνση της εξόδου μίας εντολής σε μία άλλη: **εντολή1 | εντολή2**

π.χ. `ls -l /usr/bin | grep "mk"`

Εντολή εντός άλλης εντολής: **εντολή1 $(εντολή2)**

π.χ. `echo "The current directory is $(pwd)"`

Χρησιμοποιήστε τις εντολές **history** και **grep** συνδυασμένα, για να εμφανίσετε στην οθόνη τη λίστα όλων των εντολών που χρησιμοποιήσατε για τη **δημιουργία** φακέλων

Αποθηκεύστε την έξοδο της προηγούμενης εντολής στο αρχείο **out4.log** μέσα στον φάκελο **~/test1/log**

5. Wildcards

Σύμβολο '*': ταιριάζει με οσουσδήποτε (**0 ή περισσότερους**) χαρακτήρες
Σύμβολο '?': ταιριάζει με ακριβώς **1** χαρακτήρα

Παραδείγματα χρήσης των χαρακτήρων '*' και '?' για την επιλογή πολλών "αρχείων":

/bin/mk*	όλα τα "αρχεία" που βρίσκονται στον φάκελο /bin και το όνομά τους αρχίζει από mk
*	όλα τα "αρχεία" που βρίσκονται στον τρέχοντα φάκελο
./*	όλα τα "αρχεία" που βρίσκονται στον τρέχοντα φάκελο
~/?abc?	όλα τα "αρχεία" που βρίσκονται στο home directory και το όνομά τους αποτελείται από 5 χαρακτήρες, εκ των οποίων ο 2ος, 3ος και 4ος είναι abc
~/?abc?*	όλα τα "αρχεία" που βρίσκονται στο home directory και το όνομά τους αποτελείται από 5 ή περισσότερους χαρακτήρες, εκ των οποίων ο 2ος, 3ος και 4ος είναι abc

Αντιγράψτε, χρησιμοποιώντας την εντολή **cp**, στον φάκελο **`~/test1`** όλα τα αρχεία που το όνομά τους τελειώνει σε "**.h**" και περιέχονται στον φάκελο **`/usr/include`**

Έπειτα, χρησιμοποιώντας την εντολή **mv**, μεταφέρετε από τον φάκελο **`~/test1`** τα αρχεία που αντιγράψατε στους κατάλληλους φακέλους:

m*.h → **`~/test1/include/m_files`**
s*.h → **`~/test1/include/s_files`**
υπόλοιπα ***.h** → **`~/test1/include/other_files`**

Ελέγξτε τη δομή του φακέλου με την εντολή tree και σε περίπτωση που δεν είναι σωστή χρησιμοποιήστε την εντολή mv για να κάνετε τις κατάλληλες διορθώσεις.

Αποθηκεύστε την έξοδο της εντολής **tree** για τον φάκελο **`~/test1`** στο αρχείο **`out5.log`** μέσα στον φάκελο **`~/test1/log`** .

Αποθηκεύστε την έξοδο της εντολής history στο αρχείο **`his5.log`** μέσα στον φάκελο **`~/test1/log`**

6. Ανοίξτε με τον ***emacs***[1] το αρχείο **`~/test1/test.cpp`** και γράψτε τον κώδικα του προγράμματος που ακολουθεί στην επόμενη σελίδα, μετατρέποντάς το από C σε C++, χρησιμοποιώντας τα ρεύματα εισόδου (cin) και εξόδου (cout) και τους αντίστοιχους τελεστές ">>" και "<<". Έπειτα μεταγλωττίστε το με **g++** (βλ. παράδειγμα στη διαφάνεια 11 του πρώτου μαθήματος *C++-ROOT-slides-01-Εισαγωγή.pdf*). Εφόσον αυτό γίνει επιτυχώς, θα έχει δημιουργηθεί το εκτελέσιμο αρχείο **test.exe** , το οποίο θα πρέπει να εκτελέσετε.

7. Αποθηκεύστε την έξοδο της εντολής **tree** για τον φάκελο **`~/test1`** στο αρχείο **`out7.log`** μέσα στον φάκελο **`~/test1/log`**

Αποθηκεύστε την έξοδο της εντολής history στο αρχείο **`his7.log`** μέσα στον φάκελο **`~/test1/log`**

1Μπορείτε να χρησιμοποιήσετε οποιονδήποτε διαθέσιμο επεξεργαστή απλού κειμένου επιθυμείτε, π.χ. vim, sublime, gedit κ.ο.κ.

// Πρόγραμμα C, για το 6ο ερώτημα

```c
#include<stdio.h>
#include<math.h>

int main(void) {

  double x[100], y[100], ey[100],
    S1=0, Sx=0, Sx2=0, Sy=0, Sxy=0,
    D, a, sa, b, sb;
  int i,Ndata;

  printf("Δώσε το πλήθος των μετρήσεων: ");
  scanf("%d",&Ndata);

  if (Ndata>100) {
    printf("Το πλήθος των μετρήσεων δεν μπορεί να ξεπερνά τις 100.");
    return 1;
  }

  // εισαγωγή των δεδομένων
  for (int i=0; i<Ndata; i++) {
    printf("---> %dη μέτρηση:\n",i+1);
    printf("Δώσε το x: ");
    scanf("%lf",&x[i]);
    printf("Δώσε το y: ");
    scanf("%lf",&y[i]);
    printf("Δώσε το σφάλμα στο y: ");
    scanf("%lf",&ey[i]);
  }

  // υπολογισμός των αθροισμάτων S1, Sx, Sx2, Sy, Sxy
  for (i=0; i<Ndata; i++) {
    double sigmaFactor;
    sigmaFactor=1/ey[i]/ey[i];
    S1+=sigmaFactor;
    Sx+=x[i]*sigmaFactor;
    Sx2+=x[i]*x[i]*sigmaFactor;
    Sy+=y[i]*sigmaFactor;
    Sxy+=x[i]*y[i]*sigmaFactor;
  }

  // παράγοντας D
  D=S1*Sx2-Sx*Sx;

  // εκτύπωση παραγόντων
  printf("Πλήθος μετρήσεων = %d\n",Ndata);
  printf("   Sum(1/s2)     = %f\n",S1);
  printf("   Sum(x/s2)     = %f\n",Sx);
  printf("   Sum(x2/s2)    = %f\n",Sx2);
  printf("   Sum(y/s2)     = %f\n",Sy);
  printf("   Sum(xy/s2)    = %f\n",Sxy);
  printf("   D             = %f\n",D);

  // κλίση b
  b=(S1*Sxy-Sx*Sy)/D;
  sb=sqrt(S1/D);

  // τεταγμένη επί την αρχή a
  a=(Sx2*Sy-Sx*Sxy)/D;
  sa=sqrt(Sx2/D);

  // εκτύπωση αποτελεσμάτων
  printf("Ευθεία ελαχίστων τετραγώνων: y = a + b x\n");
  printf("   τεταγμένη επί την αρχή a = %f ± %f\n",a,sa);
  printf("                    κλίση b = %f ± %f\n",b,sb);

  return 0;
}
```