BREVE SPIEGAZIONE DELLA SELEZIONE LOCALE 1999

Questo documento ha due scopi: spiegare (per chi non lo conosce) il sistema operativo UNIX che si usa durante la gara, e come si svolge la gara. In appendice c'é infine una nota per chi pensa di usare il Pascal ed é abituato a programmare in Turbo Pascal.

Questo documento é sia pubblicato on-line alla pagina ufficiale della selezione locale (http://www.disi.unige.it/person/AnconaM/ACM/), sia presentato all'inizio stesso della gara.

Introduzione a Unix

Avete a disposizione un computer con sistema operativo UNIX.
Per chi non conoscesse UNIX, molti programmi funzionano in un ambiente a finestre (alla Windows) ma i comandi vengono dati in linea come nel DOS.

Innanzitutto dovrete entrare nel computer con la login e la password che verranno assegnate.
Se si é in ambiente testuale, lanciare Xwindow con

```
startx & ; exit
```

A questo punto dovreste trovarvi di fronte un ambiente grafico con una shell dei comandi aperta. Se non ci fossero shell aperte potete aprirne una usando i menu a tendina che appaiono premendo un tasto del mouse sullo sfondo.

I comandi che si danno nella shell sono simili a quelli del DOS I comandi `cd`, `mkdir`, `rmdir` sono gli stessi del DOS. per vedere la lista dei file di un percorso usare `ls -l` al posto di `dir`.
La struttura a directory é anche essa simile, ma con tre differenze: innanzitutto non si usano i nomi dei dischi (per esempio C:), inoltre si usa "/" invece di "\". Infine, essendo UNIX un ambiente multiutente, si ha diritto di accesso completo solo in una directory (chiamata home) identificata da ~ (quindi per andare nella propria home usare `cd ~`).

Altri comandi utili sono `rm`, `cp`, `mv` che corrispondono a `del`, `copy`, `move` del DOS. Notare infine che gli eseguibili in UNIX non hanno estensione `EXE` o `COM`.

Per la gara infine servono due programmi: un editor di testo (per scrivere i programmi ed i dati di input) e un lettore di posta, con cui potrete ricevere le risposte dalla giuria, spiegazioni e classifiche parziali.

Un editor semplice é

```
textedit [nomefile] &.
```

Se non fosse disponibile chiedere a qualcun altro cosa usare al suo posto.

Come lettore di posta consiglio `elm`, non il migliore ma facile da usare.

La gara

All'inizio della gara, oltre ad una copia cartacea di questo documento, vi verrá assegnato un problema di prova: avrete circa 30 minuti per prendere confidenza con il computer, l'editor, il compilatore e la posta. Dopodiché verranno consegnati i problemi veri e propri e la gara avrá inizio. La durata della gara verrá comunicata sul momento, ma si immagina che sará di circa 3-4 ore.

Per aiutarvi sono disponibili dei file batch da utilizzare durante la gara.

Il primo va utilizzato subito, e serve per iscriversi alla gara. La sintassi é:

```
iscrivi cognome nome
```

A questo punto aprite l'editor e scrivete il sorgente del file (il nome é indicato nel testo del problema).

Una volta scritto (e salvato!) il sorgente bisognerá compilarlo.
In generale per compilare un programma di nome `nomefile.c` o `nomefile.p` (a seconda del linguaggio) usare

```
compile nomefile c [opzioni]
```

o

```
compile nomefile p [opzioni]
```

L'eseguibile si chiamerá semplicemente `nomefile`
Se volete provare a compilare in preparazione allla gara cioé quando questo batch non é disponibile usare

```
gcc nomefile.c -o nomefile [opzioni]
```

o

```
gpc nomefile.p -o nomefile [opzioni]
```

(Se `gcc` non é disponibile usare `cc`. Se `gpc` non é disponibile provare ad usare `pc`, sempre che il compilatore pascal sia disponibile sulla macchina che si sta usando. Durante la gara lo sará di sicuro.)

Se la compilazione avviene senza problemi, bisogna scrivere il file di input (indicato anche esso sul testo del problema) con i dati di prova del testo ed altri di propria invenzione e lanciare il programma. Se il programma genera l'output giusto, é il momento di mandarlo alla giuria.

Per sottomettere un sorgente alla giuria, usate il comando `submit`. La sintassi é la stessa di `compile`.

La giuria compilerá il sorgente e lancerá l'eseguibile con dei dati di prova non disponibili ai concorrenti. La risposta della giuria la riceverete via posta elettronica.

Le possibili risposte della giuria sono:
- `program accepted` il programma esegue correttamente i dati di prova.
- `wrong answer` il programma risponde con risultati sbagliati ai dati di prova.
- `compile error` la giuria non è riuscita a compilare il sorgente.
- `presentation error` il programma, sebbene corretto, non genera l'output nel formato voluto.
- `time limit exceeded` il programma non termina l'esecuzione entro pochi secondi.
- `run-time error` il programma termina con un errore.
- `contest rules violation` se il programma prova a leggere da file o canali che non siano quelli dati dal problema.

Vince la gara chi risolve piú problemi nel tempo di gara.

In caso di paritá vince chi impiega il minor tempo totale. Il tempo totale è la somma di tutti i tempi che il concorrente ha impiegato per mandare il sorgente corretto dei programmi che ha risolto dall'inizio della gara piú 20 minuti per ogni tentativo fallito di sottomissione di questi problemi.

(esempio: Pinco Pallino manda dopo un ora il sorgente corretto del problema A e dopo un'altra quello del problema B, dopo peró un tentativo andato a male. Durante tutta la gara prova a mandare piú volte sorgenti del problema C ma senza mai risolverlo. Il tempo totale sará quindi 1 ora + 2 ore + 20 minuti = 3 ore e 20 minuti)

Se durante la gare avete domande tecniche chiedete pure a voce, se invece avete domande sui problemi scrivete la domanda su un file e mandatecela con il comando

```
ask file
```

La risposta viene mandata a tutti sempre via posta elettronica.

Nota per i pascalisti

Il compilatore usato durante la gara é Pascal standard e quindi non compatibile con il Turbo Pascal della Borland. Molte funzioni e la gestione delle stringhe non sono quindi disponibili. Inoltre l'apertura dei file avviene in maniera diversa. Ecco un esempio di come avviene:

```
Program prova(fin,output);

var
  a:integer;
  ...
  fin:text;

begin
  reset (fin,´prova.in´);
  ...
  readln(fin,a);
  ...
  close (fin);
end.
```

Per commenti o richieste di spiegazioni ulteriori mandare una email a `otto@dist.unige.it` **o durante alla gara chiedere ai responsabili.**