Proposte Skull of Summer 2018!

Note

- L’ordine di **difficoltà** e **creatività** vanno da una scala da 1 (bassa) a 3 (alta).
- Tutti i lavori sono supposti essere fatti su sistemi operativi linux o mac os. In caso foste su windows, usate una macchina virtuale!
- Non è necessario completare i programmi al 100%, ci rendiamo conto che sono grossi da effettuare;
- Siamo consci del fatto che questi programmi presuppongono conoscenze che non avete. Il senso è spingervi a informarvi su degli argomenti prima ancora di programmare in modo da unire l’utile al dilettevole!

Grafi, grafi, grafi!

Difficoltà: 1
Creatività: 1

Crea una struttura dati che ti permette di rappresentare un grafo. Quindi crea un funzione che permette di generare la png di un grafo tramite il programma (dot). Dot può essere facilmente richiamato tramite la call di “system”.
Implementa quindi l’algoritmo di Kruskal e fai in modo di generare ad ogni passaggio dell’algoritmo il relativo grafo rappresentante lo stato dell’algoritmo.

Referenze utili (ma non indispensabili!)

1. https://en.wikipedia.org/wiki/Kruskal%27s_algorithm
2. http://www.graphviz.org/doc/info/attrs.html
3. https://graphviz.gitlab.io/
4. https://en.wikipedia.org/wiki/Graph_(abstract_data_type)

Goofy Internet

Difficoltà: 2
Creatività: 2

Implementa una struttura a grafo in cui ogni nodo ha una “situazione” ed una “domanda” con varie possibili “risposte”.
Implementa un CGI che ti permette, a partire da un nodo iniziale, di viaggiare in questo grafo. In particolare all’utente viene mostrata la “situazione”, la “domanda” e le possibili “risposte” del nodo in cui si trova. L’utente poi sceglie una risposta. La CGI recupera il prossimo nodo e lo mostra all’utente. In questo modo puoi creare la tua avventura testuale!
BONUS: fai in modo di poter generare il grafo da un file testuale in modo che ti sia facile aggiungere situazioni, domande e risposte.
BONUS2: fai in modo che ci siano degli stati di GOAL, ossia o di game over or di win.
BONUS3: la situazione deve essere scritta in blu, la domanda in ciano e le risposte in verde, così, giusto per.

Esempio
Situazione: “Sei in un grotta, e c’è un bivio”
Domanda: “Cosa fai?”
Scelte:

1. Vai a destra;
2. Vai a sinistra;
3. Torna da dove sei arrivato;

Referenze utili (ma non indispensabili!)

1. http://jkorpela.fi/forms/cgic.html
2. https://askubuntu.com/questions/558280/changing-colour-of-text-and-background-of-terminal

You can't see me

Difficoltà: 2
Creatività: 3

Crea un piccolo tool di steganografia che permetta di "nascondere" un messaggio di testo all'interno di un'immagine (livello 1) e di un file audio (livello 2). Sei libero/a di scegliere i formati che preferisci e il tipo di steganografia che preferisci. Noi ti consigliamo, nel caso di un'immagine, di modificare il bit meno significativo di ogni pixel per inserirci i bit del tuo messaggio.
Devi anche creare un tool che recuperi il messaggio dal file immagine.

Per esempio potresti usare delle bitmap per nascondere il tuo messaggio!

L'elettronico che c'è in te!

Difficoltà: 2
Creatività: 2

Dato il circuito:

Fare un programma che:
1. Se il pulsante #1 non è premuto, sui display a 7 segmenti sia mostrato il numero di volte in cui il pulsante #2 è stato premuto in totale. In caso di overflow, i numeri devono ritornare a 0.
2. Se il pulsante #1 è premuto mostri i numeri naturali sui display a 7 segmenti. Si parte da 0 e ne viene mostrato 1 ogni secondo;

Ma che bello, ci sono i colori

Difficoltà: 2
Creatività: 2

Dato il circuito:

Fare un programma che:
1. Accenda e spenga i LED con il pulsante #1;

2. Aumenti la velocità con cui il segnale viaggia con il pulsante #2 (velocità 0.5 LED/s; 1 LED/s, 1.5LED/s, 2.0LED/s);
3. Quando cambia il valore del potenziometro si cambia anche il tipo di segnale mostrato. Butto giù degli esempi: y=sin(x), y=sin(2x), y=e^(x%100)*sin(0.05x)

ASCII art

Difficoltà: 3
Creatività: 1

Dato il circuito:

Fare in modo che:

1. Il tuo programma riesca ad ottenere i valori di accelerazione x,y,z in un dato momento;
2. Il tuo programma riesca ad ottenere i valori di angolazione del giroscopio roll, pitch,yaw in un dato momento;
3. Se il valore di roll è basso, illuminare il LED 1;
4. Se il valore di roll è alto, illuminare il LED 2;
5. Se il valore di pitch è basso, illuminare il LED 3;
6. Se il valore di pitch è alto, illuminare il LED 4;
7. Se il valore di yaw è basso, illuminare il LED 5;
8. Se il valore di yaw è alto, illuminare il LED 6;
9. Il programma deve mostrare, in **realtime**, un istogramma in un cui ci sono i valori di accelerazione x, y, z, roll, pitch, yaw. Per esempio:

Potreste voler collaborare con il team “Gif or Jif?” per questa cosa!
PS: sì il nome ti ha ingannato! Pensavi che non ci fosse un circuito nhé?

Forza 4!!!

Difficoltà: 1
Creatività: 1
Quando il gioco si fa duro, spegni la console e lasciala raffreddare.

Implementare un forza 4; le regole sono:

- Da 2 a 4 giocatori (ogni giocatore ha un nome);
- Ogni turno un giocatore mette un gettone su una colonna della tavola a sua scelta;
- I gettoni cadono fino al terreno o fino a che non trovano un altro gettone;
- Il primo giocatore che mette in fila 4 gettoni vince;

Ad ogni turno deve essere mostrato, in console, lo stato del gioco colorato nel seguente modo:

```
| | | | | | | | | | |
-----------------
| | | | | | | | | | |
-----------------
```

Questa griglia poi può essere colorata con degli spazi vuoti nel seguente modo:

- Nero: casella disponibile;
- Rosso: casella con un gettone del giocatore 1;
- Giallo: casella con un gettone del giocatore 2;
- Verde: casella con un gettone del giocatore 3;
- Blu: casella con un gettone del giocatore 4;

Bonus: invece di usare standard ASCII, puoi usare l'unicode poiché ti permette di disegnare una griglia esteticamente migliore!

BONUS: puoi inserire anche delle "intelligenze" artificiali che inseriscono i propri gettoni in colonne perfettamente casuali! Ah, dove ci porterà il progresso!

Referenze (non necessariamente utili)

1. https://en.wikipedia.org/wiki/Box-drawing_character
2. https://askubuntu.com/questions/558280/changing-colour-of-text-and-background-of-terminal

Michele, ti voglio bene!

Difficoltà: 1
Creatività: 2

Nella scorsa Skull of Summer, un partecipante, Michele, dovette scrivere un libreria per gestire **liste, heap e grafi**.

Purtroppo non è riuscito a finire la cosa, quindi ora tocca a te! Il tuo obiettivo è finalizzare il lavoro di Michele, testarlo, generare un **libreria dinamica** in modo tale che il tuo progetto possa essere utilizzato dagli **altri team** per finalizzare il loro di lavoro. Il tempo è ancora più scarso per te!

Di seguito alcune (INUTILI) informazioni su liste, heap e grafi:

- La lista è una concatenazione di nodi, in cui ciascun nodo segue il precedente, ad eccezione del primo nodo (la testa) che non è preceduto da nessuno, mentre per l'ultimo nodo (la coda) non esiste alcun successore. Una lista, normalmente, viene definiti unidirezionale (è possibile scorrere in un sol senso). Talvolta è definita come bidirezionale, pertanto da ciascun nodo è possibile risalire a chi lo precede o recuperare il nodo successivo.
- L'HEAP è una struttura ad albero, ove ciascun nodo è dotato di una propria chiave (genericamente numerica, ma non è necessariamente), in cui è valida la "proprietà dell'heap": in cui le chiavi sono vincolate da un ordine preciso. Ne consegue che, generalmente, esistono alberi di max/min heap in cui il nodo padre possiede sempre la chiave max/min rispetto ai nodi figli.
- Il Grafo è concettualmente una lista in in cui ciascun elemento possiede più di un collegamento con altri pari. Un set definito di nodi collegati tra loro, costituiscono un grafo se e solo se è possibile partire da uno qualsiasi e raggiungerne un altro, in un numero finito di passi.

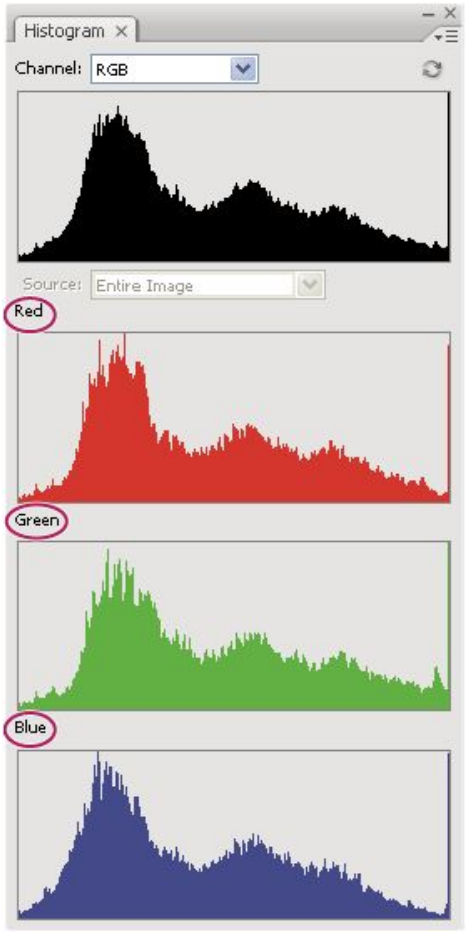
La liberia deve poter gestire le note operazioni di add/remove/get dei nodi, a seconda delle scelte fatte dal programmatore.

Il progetto è disponibile nella cartella:
https://github.com/STB1019/SkullOfSummer/tree/master/pjt/2017/realizzazioni/SkullOfSummer_ProgettoDusi

GIF o JIF?

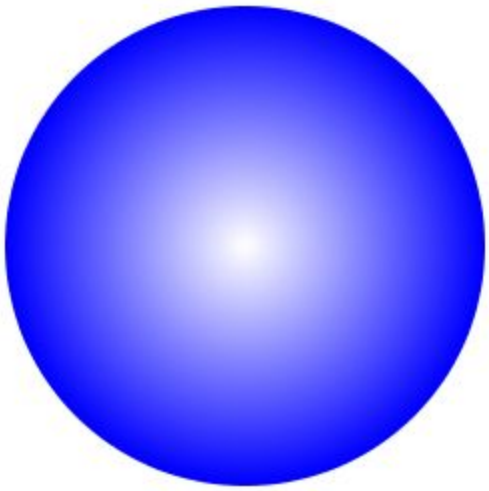
Il tuo compito è creare un programma che legga della gif e mostra l'istogramma dei colori RGB al suo interno. Se la gif è animata, l'istogramma deve mostrare i valori medi (consiglio: all'inizio non sceglirne una animata!). Implementa un lettore di GIF (o JIF!) minimale, senza usare le proprietà opzionali del formato.

Per intenderci, l'istogramma è un grafico in cui, sulle X ci sono valori da 0 a 255 e sulla Y c'è un valore che ti dice quanti pixel hanno quel particolare colore R, G o B.
In pratica tu dovrai avere 3 istogrammi nella gif. Una cosa del tipo:



Ma fatta come:

Potreste voler collaborare con il team “ASCII art” per questa cosa!
BONUS: il programma ora non solo può mostrare un istogramma ma ti permette di creare automaticamente una gif in cui c’è un cerchio con un colore in gradiente:



Il colore lo potete scegliere voi.

Link non necessariamente utili:

- http://giflib.sourceforge.net/whatsinagif/bits_and_bytes.html

Il rumore bianco più importante del secolo

Difficoltà: 2;
Creatività: 2;

Considera il software tcpdump che ti permette di leggere cosa sta leggendo il tuocomputer su una porta TCP.
Il tuo compito è quello di leggere i valori che escono dalla porta 80, convertirli in integer a 8 byte e trarne delle statistiche.
Per statistiche intendiamo creare un file CSV che contenga i valori rilevati;
Su stdout, deve stampare la media attuale dei valori e la relativa varianza. Una volta soddisfatti, l'utente deve premere un tasto: dopo la pressione del tasto, il programma dovrà richiamare **gnuplot** e generare automaticamente un grafico dell'andamento del segnale ricevuto.

```
# tcpdump -XX -i eth0
11:51:18.974360 IP 172.16.25.126.ssh > 172.16.25.125.apwi-rxspooler: Flags [P.], seq 3509235537:3
0x0000:  b8ac 6f2e 57b3 0001 6c99 1468 0800 4510  ..o.W...l..h..E.
0x0010:  00ec 8783 4000 4006 275d ac10 197e ac10  ....@.@.']...~..
0x0020:  197d 0016 1129 d12a af51 d9b6 d5ee 5018  .}...).*.Q....P.
0x0030:  4948 8bfa 0000 0e12 ea4d 22d1 67c0 f123  IH.......M".g..#
0x0040:  9013 8f68 aa70 29f3 2efc c512 5660 4fe8  ...h.p).....V`O.
0x0050:  590a d631 f939 dd06 e36a 69ed cac2 95b6  Y..1.9...ji.....
0x0060:  f8ba b42a 344b 8e56 a5c4 b3a2 ed82 c3a1  ...*4K.V........
0x0070:  80c8 7980 11ac 9bd7 5b01 18d5 8180 4536  ..y.....[.....E6
0x0080:  30fd 4f6d 4190 f66f 2e24 e877 ed23 8eb0  0.OmA..o.$.w.#..
0x0090:  5a1d f3ec 4be4 e0fb 8553 7c85 17d9 866f  Z...K....S|....o
0x00a0:  c279 0d9c 8f9d 445b 7b01 81eb 1b63 7f12  .y....D[{....c..
0x00b0:  71b3 1357 52c7 cf00 95c6 c9f6 63b1 ca51  q..WR.......c..Q
0x00c0:  0ac6 456e 0620 38e6 10cb 6139 fb2a a756  ..En..8...a9.*.V
0x00d0:  37d6 c5f3 f5f3 d8e8 3316 d14f d7ab fd93  7.......3..O....
0x00e0:  1137 61c1 6a5c b4d1 ddda 380a f782 d983  .7a.j\....8.....
0x00f0:  62ff a5a9 bb39 4f80 668a                 b....9O.f.
11:51:18.974759 IP 172.16.25.126.60952 > mddc-01.midcorp.mid-day.com.domain: 14620+ PTR? 125.25.1
0x0000:  0014 5e67 261d 0001 6c99 1468 0800 4500  ..^g&...l..h..E.
0x0010:  0048 5a83 4000 4011 5e25 ac10 197e ac10  .HZ.@.@.^%...~..
0x0020:  105e ee18 0035 0034 8242 391c 0100 0001  .^...5.4.B9.....
0x0030:  0000 0000 0000 0331 3235 0232 3502 3136  .......125.25.16
0x0040:  0331 3732 0769 6e2d 6164 6472 0461 7270  .172.in-addr.arp
0x0050:  6100 000c 0001                           a.....
```

Questo è l'output di tcp che ci apettiamo. I dati che devi prendere sono i vari valori esadecimali (spazi esclusi) che stanno tra la seconda colonna (inclusa) fno alla nona colonna (inclusa). Ognuno di questi caratteri deve essere poi converrtito come precedentemente specificato.
Il comando di tcpdump da usare dovrebbe essere: "tcpdump -c <numeropacchetti> -XX -i <tuainterfaccia> port 80". Siccome questa chiamata è bloccante devi fare una statistica su n pacchetti ricevuti. Il numero "n" deve essere scelto dall'utente.

Progetto keylogger mancato

Difficoltà: 1;
Creatività: 1;

Crea un programma che lancia in background un thread in cui legge tutto ciò che batti nella tastiera. Questo programma pubblica delle statistiche riguardo il tuo battito di tastiera nel file “/proc/keyboard/local/statistics”. Questo percorso è un percorso del tutto fittizio in linux!

Le statistiche da considerare sono:

1) Numero di caratteri battuti;
2) Battiti al secondo (il timer della statistica viene lanciata al primo tasto premuto e termina automaticamente se non vengono battuti più tasti per 3 secondi);
3) Tasto più premuto;
4) Parola più battuta (una parola, per definizione, è quella stringa di caratteri 0-9,A-Z,a-z compresi tra 2 caratteri non appartenenti ad una parola; per esempio in “...ciao….mondo!!!” “ciao” e “mondo” sono 2 parola);
5) Tempo analizzato;