



Cognome	Nome	Matricola	Voto: ... /30
----------------	-------------	------------------	----------------------

Quesito:	1	2	3	4	5	Tot.
Max:	6	7	2	10	5	30
Punti:						

Istruzioni:

- non è possibile consultare libri, appunti, né comunicare;
- non è possibile utilizzare la calcolatrice o qualsiasi dispositivo elettronico;
- si può rispondere ai quesiti nell'ordine preferito;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del **rosso**.
- tempo a disposizione: 1h 45m

Stile del codice C:

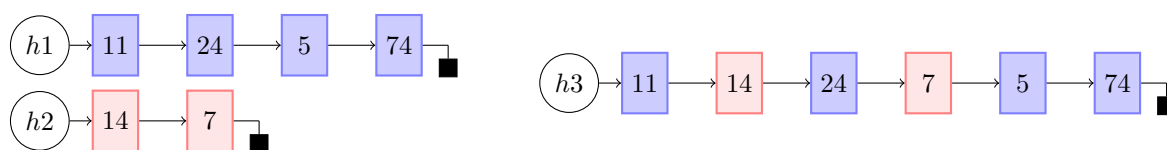
- non è necessario inserire direttive #include;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è consentito l'utilizzo di funzioni di libreria;

Quesito 1 (6 punti)

Punteggio ottenuto: ... /6

Si scriva un sottoprogramma che riceve in ingresso due teste di liste per la gestione di numeri interi ne crei e restituisca una nuova costituita dagli elementi delle liste di partenza presi alternativamente dall'una e dall'altra lista. Le liste iniziali devono rimanere invariate.

Esempio:



Quesito 2 (7 punti)

Punteggio ottenuto: ... /7

Scrivere un sottoprogramma che ricevuta in ingresso una stringa contenente una frase, crei una seconda stringa in cui le parole che compongono la frase sono in ordine inverso e la restituisca. Si ipotizzi di avere a disposizione un sottoprogramma `strsplit` il cui prototipo è: `strlist * strsplit(char * src, char sep);` che ricevuta in ingresso una stringa `src` restituisce una lista in cui ciascun elemento contiene una sottoparte della stringa, delimitata da `sep`. Si consideri inoltre dato il sottoprogramma che riceve in ingresso una lista del tipo prima utilizzato, la restituisce dopo aver invertito l'ordine dei suoi elementi, il cui prototipo è: `strlist * listinvert(strlist *);`

Esempio:

Ingresso: questo esercizio richiede una ventina di righe di codice

Uscita: codice di righe di ventina una richiede esercizio questo

Quesito 3 (2 punti)

Punteggio ottenuto: ... /2

Scrivere un sottoprogramma che ricevuta in ingresso una lista del tipo prima utilizzato, la restituisce dopo aver invertito l'ordine dei suoi elementi (non è una nuova lista).

Quesito 4 (10 punti)

Punteggio ottenuto: ... /10

Scrivere un programma per la verifica della vincita del superenalotto. Il programma acquisisce i 6 numeri interi estratti, il numero jolly, ed il numero superstar. A questo punto, il programma chiede ciclicamente i 6 numeri giocati, l'eventuale numero superstar (l'utente mette -1 se non l'ha giocato) ed il programma visualizza l'eventuale combinazione vincente:

- punti 6: indovinati 6 numeri
- punti 5+: indovinati 5 numeri e il jolly
- punti 5: indovinati 5 numeri
- punti 4: indovinati 4 numeri
- punti 3: indovinati 3 numeri
- 5 stella: indovinati 5 numeri e numero star
- 4 stella: indovinati 4 numeri e numero star
- 3 stella: indovinati 3 numeri e numero star
- 2 stella: indovinati 2 numeri e numero star
- 1 stella: indovinati 1 numeri e numero star
- 0 stella: indovinati 0 numeri e numero star

Visualizzato il risultato dell'analisi, il programma chiede se continuare con la verifica di un'altra sequenza (1) oppure terminare (0).

Esempio:

```
Ingresso: 8 26 33 40 62 88      87 18
Ingresso: 1 3 28 30 40 78      -1
Uscita:  nessuna vincita - continua (1 si | 0 no)?
Ingresso: 1
Ingresso: 31 33 50 62 65 80      18
Uscita:  2 stella - continua (1 si | 0 no)?
Ingresso: 0
Uscita:  termine.
```

I numeri, sia quelli estratti, sia quelli giocati, sono forniti già ordinati in senso crescente. Non ci sono ripetizioni. Si suggerisce di strutturare il programma in sottoprogrammi (per esempio `leggiNumeri` che acquisisca i sei numeri e li metta in un array, `quantiNumeri` che riceva due array e qualsiasi altro parametro necessario, restituisca il numero di valori comuni ai due array ...).

Quesito 5 (5 punti)

Punteggio ottenuto: ... /5

Si consideri il programma di seguito riportato:

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    printf("b\n");
    if(fork()==0){
        printf("a\n");
        fork();
        printf("c\n");
    } else
        wait();

    printf("d\n");
}
```

Rispondere alle seguenti domande (**direttamente su questo foglio**).

1. quanti processi vengono creati (incluso il processo principale)?
2. cosa viene prodotto in uscita?
3. è univoca l'uscita prodotta? perchè?