

Cognome:		Nome:	
Matricola:		Anno immatricolazione:	
<input type="checkbox"/> 1° Parziale (30min) ESERCIZI dal 1 al 6	<input type="checkbox"/> 2° Parziale (30min) ESERCIZI dal 7 al 9	<input type="checkbox"/> Totale 6 CFU (1h) ESERCIZI dal 1 al 9	
<p>PROPEDEUTICITÀ: lo studente è consapevole che NON PUÒ SOSTENERE questo esame SE NON SI È SUPERATO l'esame di "Laboratorio di Programmazione di Sistema - LPS" (precedentemente "Laboratorio di Architettura degli Elaboratori - LAE")</p> <p><input type="checkbox"/> Lo studente dichiara di aver già superato l'esame di LPS (LAE).</p> <p><input type="checkbox"/> Lo studente dichiara di voler sostenere questo esame CON RISERVA SUL SUPERAMENTO DI LPS IN UNO DEGLI APPELLI DELLA SESSIONE CORRENTE. Lo studente è consapevole che PERDERA' IL VOTO DI QUESTO ESAME SE NON SUPERERA' LPS IN UNO DEGLI APPELLI DELLA SESSIONE CORRENTE.</p>			

- Domande a risposta multipla → **2 punti** per ciascuna risposta esatta, **-1 punti** per ciascuna risposta errata, **0 punti** per ogni risposta omessa. Le domande a risposta multipla possono avere una e una sola risposta esatta.
- Per indicare la risposta scelta cambiare il **colore del testo in rosso**, oppure sottolineare, oppure **evidenziare**.

1. Il comando ls -le pippo

- stampa tutte le linee del file pippo che contengono la stringa "le"
- stampa l'Access Control List (ACL) associata con il file pippo, se presente
- stampa l'elenco di tutti gli extended attribute del file pippo che contengono la stringa "le"

2. Siano "16 -rwxr-xr-x@ 1 marcoautili staff 4631 Jan 16 2020 apue.h" le informazioni stampate dal comando "ls -las apue.h", il numero 16 più a sinistra indica

- il numero di blocchi occupati nel disco dal file apue.h
- che il file apue.h ha 16 attributi estesi ad esso associati
- l'id del superuser del file apue.h

3. La pipeline ls -l | grep rwxrwxrwx

- stampa l'elenco di tutti i file nella directory corrente con permessi rwxrwxrwx
- stampa l'elenco di tutti i file nella directory root con permessi rwxrwxrwx
- stampa l'elenco di tutti i file nella directory corrente con permessi rwxrwxrwx o che hanno rwxrwxrwx nel loro nome

4. Il comando sed -e 's/&/&/g' -e 's/&/&/g' < p.htm

- stampa su stdout il contenuto del file p.htm sostituendo tutte le occorrenze di & con #38;
- stampa su stdout il contenuto del file p.htm sostituendo tutte le occorrenze di & con & e #38; con &
- stampa su stdout il contenuto del file p.htm sostituendo tutte le occorrenze di & con #38;

5. Scrivere nelle due righe a destra cosa stampa il seguente script.

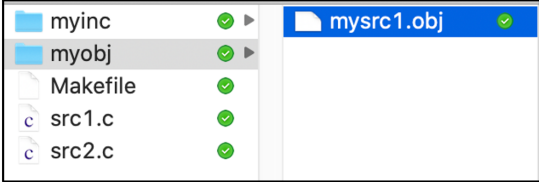
es5.sh	
<pre>#!/bin/bash v1='castagne' function func () { v1="\$1 <u>castagne</u> and \$2" echo "\$v1" } v2="\$(func pasta ceci) <u>mixed</u>" echo "<u>only</u> \$v1" echo "\$v2 <u>together</u>"</pre>	<p style="text-align: center;"><u>Only castagne</u></p> <p style="text-align: center;"><u>pasta castagne and ceci mixed together</u></p>

6. Descrivere nella riga sotto cosa fa il seguente comando.

```
sed -e 's/^/ /g' pippo.txt
```

Sostituisce l'inizio di ogni riga con lo spazio nel file pippo.txt

7. Considerare i seguenti file e la loro disposizione nella directory corrente *ESERCIZIO7*, all'interno della quale: la sottodirectory *myobj* contiene solo il file *mysrc1.obj* e la sottodirectory *myinc* contiene il file *myheader.h*.

ESERCIZIO7		Makefile
		<pre>myobj/my%.obj: %.c myinc/myheader.h gcc -c -o \$@ \$< -I./myinc main1: myobj/mysrc1.obj myobj/mysrc2.obj gcc -o \$@ \$^ -I./myinc .PHONY: clean clean: rm -f myobj/my*.obj *~ core ./myinc/*~</pre>
src1.c	src2.c	./myinc/myheader.h
<pre>#include <myheader.h> int main() { myOPSLabExam(); return(0); }</pre>	<pre>#include <stdio.h> #include <myheader.h> void myOPSLabExam(void) { printf("It's quite easy!\n"); return; }</pre>	<pre>void myOPSLabExam(void);</pre>

Assumendo che il file *src1.c* non abbia subito modifiche dall'ultima compilazione andata a buon fine senza errori, scrivere nelle righe sotto (non usare tutte le righe) qual è l'output sul terminale dell'esecuzione del comando *make*:

```
gcc -c -o myobj/mysrc2.obj src2.c -I./myinc
gcc -o main1 myobj/mysrc1.obj myobj/mysrc2.obj -I./myinc
```

INOLTRE, scrivere nelle righe sotto (non usare tutte le righe) qual è l'output sul terminale di un'ulteriore esecuzione del comando *make* dopo aver modificato il file *myheader.h*.

```
gcc -c -o myobj/mysrc1.obj src1.c -I./myinc
gcc -c -o myobj/mysrc2.obj src2.c -I./myinc
gcc -o main1 myobj/mysrc1.obj myobj/mysrc2.obj -I./myinc
```

8. Descrivere nella riga sotto cosa fa il seguente programma.

```
#include "apue.h"
#define PROGPATHNAME "${PROGPATHNAME:-head -n 8}"

int
main(int argc, char *argv[])
{
    char    line[MAXLINE];
    FILE    *fpin, *fpout;

    if (argc != 2)
        err_quit("usage: es8 <pathname>");

    if ((fpin = fopen(argv[1], "r")) == NULL)
        err_sys("can't open %s", argv[1]);

    if ((fpout = popen(PROGPATHNAME, "w")) == NULL)
        err_sys("popen error");

    while (fgets(line, MAXLINE, fpin) != NULL) {
        if (fputs(line, fpout) == EOF)
            err_sys("fputs error to pipe");
    }

    if (ferror(fpin))
        err_sys("fgets error");
    if (pclose(fpout) == -1)
        err_sys("pclose error");

    exit(0);
}
```

Stampa le prime 8 righe del file

9. Completare il seguente programma C che legge dallo standard input dei numeri e ne stampa la somma nello standard output come segue: "IL TOTALE E': 256"

```
#include ...

int main(int argc, char *argv[]) {
    int n [30], totale _____;

    printf("Inserisci una sequenza di numeri interi separati da [INVIO].\n");
    printf("Terminare la sequenza con [Ctrl-D] \n");

    while( scanf("%d", n) != EOF _____ ) {
        totale+=n;
    }

    printf("IL TOTALE E': %d \n", totale) _____;
}
```