- **Se si sta svolgendo l'esame a distanza, per indicare la risposta scelta cambiare il colore del testo in rosso, oppure sottolineare, oppure evidenziare.**

1. **Indicare quale dei seguenti comandi consente di modificare i permessi del file "pippo" dalla maschera r-- -w- r-x alla maschera rw- r-- rw-**
   a. chmod u+w pippo; chmod a+x pippo
   b. chmod 642 pippo ; chmod o-w pippo
   c. chmod 642 pippo ; chmod o+r pippo

2. **Il comando file ./\!e* | while read A B ; do echo $B ; done stampa:**
   a. il nome seguito dal tipo di tutti i file contenuti nelle directory i cui nomi iniziano con "!e"
   b. il solo tipo di tutti i file nella directory corrente i cui nomi iniziano con "!e"
   c. il nome seguito dal tipo di tutti i file nella directory corrente i cui nomi finiscono con "e*"

3. **Il comando chmod -N <some file>**
   a. stampa la lista di permessi Nulli (non validi) associati al file <some file>
   b. rimuove i permessi Nulli (non validi) associati al file <some file>
   c. rimuove l'Access Control List associata al file <some file>

4. **Il comando chmod +a "user:pluto deny write" pippo**
   a. cambia i permessi del file pippo per permettere l'acceso all'utente pluto
   b. cambia la entry ACL per togliere i permessi di scrittura all'utente pluto e darli all'utente pippo
   c. aggiunge una entry ACL al file pippo per revocare il permesso di scrittura all'utente pluto

5. **Scrivere nella sola riga in basso a destra cosa stampa il comando es5.sh 6**

18:26

Richiedi controllo

1 PARZ A - 2021-11-12

**5. Scrivere nella sola riga in basso a destra cosa stampa il comando es5.sh 6**

**es5.sh**

```
#!/mybin/bash

function myfunction {

    if [ $1 -gt 1 ]; then
        ((a=$1-1))
        ((b=$1-2))
        temp1=$(myfunction $a)
        temp2=$(myfunction $b)
        ((res=temp1+temp2))
    else
        if [ $1 -eq 1 ]; then
            res=1
        else
            res=0
        fi
    fi
    echo $res
}

res=$(myfunction $1)
echo $res
```

8



Marco Autili

6. **Scrivere nelle due righe sotto la regola che un Makefile può contenere per rimuovere eventuali file di backup creati durante la fase di compilazione**

clean: rm -rf *o hello

7. **Sapendo che la funzione "preprocessing ()" cancella ogni occorrenza di "<" e sostituisce ogni occorrenza di ">" e "/" con ";" scrivere nelle tre righe sotto cosa stampa il comando ./es7.sh es7.xml**

**es7.sh**

```
#!/mybin/bash

xmlgetnext () {
    local IFS=';'
    read TAG VALUE TAG
}

pre_processing () {
    ...
}

cat $1 | pre_processing | sort -r | while xmlgetnext ;
do echo "$TAG $((++CONT)) = $VALUE"
done
```

**es7.xml**

```
<data>topolino 4</data>
<data>minnie 16</data>
<data>pluto 2</data>
```

data 1 = topolino 4
data 2 = pluto 2
data 3 = minnie 16

8. **Scrivere nelle quattro righe sotto, quali sono le fasi principali dell'intero processo di compilazione C studiate in classe**

preprocessing
compilation
assemblyng
linking

9. **Assumendo che il Makefile seguente sia l'unico file contenuto nella directory corrente, scrivere nelle righe sotto (non usare tutte le righe) qual è l'output sul terminale dell'esecuzione del comando *make***

Makefile

```
pluto: pippo.o
	cc $^ -o $@

pippo.o: pippo.c
	cc -c pippo.c -o pippo.o

pippo.c: minnie.c
	echo "int main() { return 0; }" > pippo.c

minnie.c:
	echo "CIAO CIAO"
```

echo "ciao ciao"(oppure ciao ciao)
echo "int main() {return o;}">pipp.c
cc -c pippo.c -o pippo.o
cc pippo.o -o pluto