

Nome e Cognome: _____

Matricola: _____

1. **6 punti** Scrivere cosa stampa il seguente programma.

```
1 int a= 0xfe - 067;
2 printf("%d\n", a);
3 while(a > 9 ? !!!0: !(a+1, a-=2, --a)) {
4     printf("%d \n", a);
5     if (a + 2> 0x11) {
6         a= -3;
7         continue;
8     }
9     a+= 4;
10 }
11 !(a+1) || a++;
12 printf("a: %d\n", a);
```

199
199
-6
-5
-4
-3
-2
-1
a: 1

2. **5 punti** Su foglio protocollo, elencare e descrivere le parti in cui è diviso *gcc*.
3. **6 punti** Data la seguente *struct* scrivere la definizione una funzione di nome *inserisci_elemento* che prende come parametro una posizione *i* ed un valore intero *n*, ed inserisce *n* in posizione *i* (si supponga la lista come globale): ad esempio, se la lista è 7-1-2, *i* == 2, *n* == 3, la lista finale sarà 7-3-1-2.

```
1 struct Node {
2     int info;
3     struct Node* pNext;
4 }
```

4. **6 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -o main main.c*, 2) *gcc -o write write.c*, 3) *gcc write.c main.c -o main*. In caso il punto 3) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno *count*, *i*, e *mywrite*, ed in quale file sono definite? Cosa stampa il programma?

<p>main.c</p> <pre>void mywrite(int *count); extern int count; int main(void) { do { mywrite(&count); } while(count <= 0); }</pre>	<p>write.c</p> <pre>#include <stdio.h> int i = 1; static int count = -4; void mywrite(int *a) { static int count= -3; (*a)++; printf("%d\n", count= count + i); }</pre>
---	---

1) errore linker: mancano le definizioni di count e mywrite
2) errore linker: manca la definizione di main
3) errore linker: la definizione di count non è visibile in main.c, si può correggere eliminando la parola chiave "static" nella sua definizione in write.c

Dopo aver corretto l'errore, count e i globali sono definiti in write.c ed hanno linkage esterno, mywrite è definita in write.c ed ha linkage esterno, count locale alla funzione mywrite in write.c ha no linkage (ed è definita in write.c)

Il programma stampa
-2
-1
0
1
2

5. **7 punti** Cerchiare le affermazioni vere dato $\text{long long } a[3] = \{1536, -2, \text{LLONG_MIN} + 512\};$
 $\text{short int } *p = (\text{short}*) a; \text{ char } *q = (\text{char}*) a; p[1] = 4097, p[3] = 4095-2, *(q+15) = 73, p[9] = 4096*4+1;$
sapendo che i tre tipi usati occupano 8, 2, e 1 byte, e $4096 = 2^{12}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria, cerchiare e giustificare le affermazioni (vere o false). Gli operatori *|* e *&* ritornano rispettivamente l'*or* e l'*and* bit-a-bit dei due operandi, mentre *<<* rappresenta l'operatore di *shift* di *n* posizioni a sinistra.

A. $(q[8] | q[2]) + q[2]$ B. $((\text{int})(p + 11) - (\text{int})(a + 2)) + q[18] \% 7$ C. $((q[18] << 1) \& (q[19])) == (q[20] >> 5)$

Esercizio 2:

Gcc è diviso in 4 parti: preprocessor, compiler, assembler e linker (descrizione su slide)

Slide: https://francescosantini.sites.dmi.unipg.it/slides/progl/20_compiler.pdf

<https://medium.com/@gpradinett/the-four-stages-of-the-gcc-compiler-preprocessor-compiler-assembler-linker-3dec8714bb9c>

Esercizio 3

Supponiamo il puntatore ad inizio lista globale pHead

```
int inserisciElemento(int n, unsigned int i)
{
    int ritorno = -1;
    NodePointer pTmpNode = NULL;
    unsigned int iCount = 0;

    pTmpNode = pHead;

    for( iCount = 1; ((iCount < i) && (pTmpNode!= NULL)) ; iCount++)
    {
        pTmpNode = pTmpNode ->pNext;
    }

    if (pTmpNode == NULL)
    {
        printf("Posizione invalida\n");
        return ritorno;
    }
    else
    {
        struct Node* pNewNode = malloc(sizeof(struct Node));
        if( pNewNode != NULL)
        {
            pNewNode->info = n;
            pNewNode->pNext = pTmpNode->pNext;
            pTmpNode->pNext = pNewNode;
            ritorno = 0;
        }
    }
    return ritorno;
}
```

00000000
 01100000
 10000000 ← q[2]
 00001000 ← q[3]
 &p[2]

00000000
 00000000
 10111111
 11110000

p[4]
 q[8]
 *(p+5)

11111111
 11111111
 11111111
 10010010

a+2
 p[9]
 q[18]

00000000
 00000000
 00000000 ← p+11
 00000001

$$A \quad (q[8] \mid q[2]) + q[2] == 0$$

11111111 1 FALSA
 == -1

$$B \quad (int)(p+1) - (int)(a+2) + q[18] > 7 == 0$$

6 + 1 FALSA
 ==

C $q[18] \in q[20]$
 IN BIG ENDIAN

$$q[18] \rightarrow 00000001 >> 1 \quad q[20] \rightarrow 00000000 >> 5$$

== ==
 00000000 & q[18] 00000000
 == =
 0 = 0
 1 VERA