

Nome e Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_

1. 3 punti Elencare le conversioni di tipo (... da ... a). Dato  $USHRT\_MAX = 65535$ , scrivere il valore finale della variabile  $b$  sapendo che il carattere  $a$  ha valore 97 e le altre lettere seguono in ordine alfabetico.

```

1 int fun2(int p) {
2     return (p + 'd'); }
3
4 long int fun1(long int p) {
5     char a= 'a';
6     return fun2(p - a); }
7
8 int main(void) {
9     unsigned short a = -4;
10    double b= fun1(a);
11 }
```

per la regola (slide 18 conversioni)

**linea 9: -4 da int a unsigned short**  
**linea 10: a da unsigned short a long int**  
**linea 5: 'a' da int a char**  
**linea 6: a da char a long int in p-a**  
**linea 6: p-a da long int a int in chiamata fun2**  
**linea 6: risultato fun2 da int a long int (ritorno fun1)**  
**linea 10: risultato fun1 da int a double**

$n + (USHRT\_MAX + 1)$   
**ad a viene assegnato**  
 $-4 + (USHRT\_MAX + 1) = 65532$   
 $b = 65532 - 97 + 100 = 65535.0$

2. 3 punti Scrivere cosa stampa la seguente porzione di codice.

```

1 int main() {
2     int i, s, c= 0, c1= 0, a= 4, k=0;
3     for(i=1; i<=a; ++i, k=0) {
4         for(s=1; s<=a-i; ++s) {
5             printf("-");
6             ++c; }
7         while(k != 2*i-1) {
8             if (c < a-1) {
9                 printf("*-"); ++c; continue;}
10            else {
11                ++c1; printf("%d-", (i+k-1*c1));}
12            ++k; }
13        c1 = c = k = 0;
14        printf("\n"); }
15    return 0; }
```

```

--0-
--*-1-1-1-
--*-2-2-2-2-
--*-3-3-3-3-3-3-
```

3. 3 punti Scrivere cosa stampa la seguente porzione di codice.

```

1 int a= 0xfb;
2 while(a > 021 ? !!1: !(a--, --a)) {
3     printf("%d \n", a + 1);
4     if (a + 2 >= 0x11) {
5         a= -0x1;
6         continue; }
7     a+= 3; }
8 a+1 || a++;
9 printf("a: %d\n", a);
```

```

252
-2
-1
0
a: 0
```

4. 4 punti Su foglio protocollo, scrivere la definizione di una funzione *sum\_diagonals* che somma gli elementi sulla prima diagonale (*somma\_d1*) e quelli sulla seconda diagonale (*somma\_d2*) di una matrice  $n \times n$  (matrice di *int*, ricevuta come parametro dalla funzione) e ritorna 1 se  $somma\_d1 == somma\_d2$ , o 0 altrimenti.
5. 3 punti Data la seguente struttura ed un puntatore ad inizio lista *struct Node \*pFirst*, definire una funzione di nome *print\_l* che stampa su video il valore del campo *info* per tutti i primi  $l$  elementi della lista. Se  $l == 3$  e la lista è lunga 5, il quarto e quinto elemento non vengono considerati.

#### Esercizio 4

```
int sum_diagonals (int n, matrix[n][n]) {
    int somma_d1, somma_d2;
    for (int i= 0; i < n; i++) {
        somma_d1+= m[i][i];
        somma_d2+= m[i][n-i-1];
    }
    return (somma_d1== somma_d2);
}
```

#### Esercizio 5

```
void print_l(Node* pFirst, int l) {
    if(pFirst == NULL)
        printf("Lista vuota!");
    else {
        if (l > 0) {
            Node* pScan = pFirst;
            l-= 1;
            do{
                printf("Info: %d\n", pScan->info);
                pScan = pScan->pNext;
            }while(pScan!= NULL && l > 0);
        }
    }
    return;
}
```

#### Esercizio 6

```
int sumDigits(int num) {
    static int sum=0;

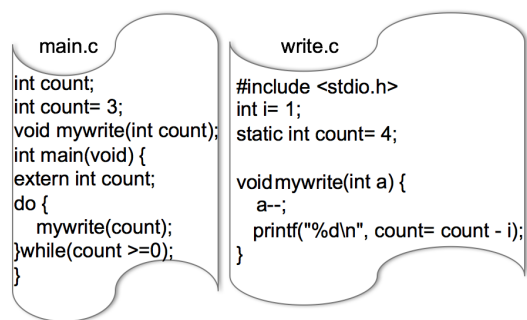
    if(num>0) {
        sum+= num%10;
        sumDigits(num/10);
    }
    else
        return sum;
}
```

```

1 struct Node {
2     int info= 0;
3     struct Node* pNext= NULL;
4 }

```

6. **3 punti** Su foglio protocollo, scrivere una funzione ricorsiva di nome *digit\_sum* che calcola la somma di tutte le cifre di un numero intero passato come input. Per esempio, se l'input è 1232 il risultato deve essere 8. Tutto il codice deve esser contenuto all'interno della funzione. Quanti *frame* vengono aperti se l'input è 1232?
7. **4 punti** Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -c main.c*, 2) *gcc -o write write.c*, 3) *gcc write.c main.c -o main*, 4) *gcc -o main main.c*. In caso il punto 3) ritorni un errore, descrivere come può essere corretto. Infine, che tipo di *linkage* hanno *count*, *i*, e *mywrite*? Cosa stampa il programma *output*? Elencare tutte le definizioni e dichiarazioni in ogni file.



2) errore: manca la definizione di main	Stampa
4) errore: manca la definizione di mywrite	3
count ha linkage esterno in main.c, interno in write.c	2
i ha linkage esterno in write.c	1
mywrite ha linkage esterno in main.c e write.c	0
	-1
	e continua
in main.c: primo count (globale) è un tentativo di definizione (rimane dichiarazione), secondo count globale è definita, mywrite dichiarata, count (locale) è dichiarata.	
in write.c: i è definita, count è definita, mywrite è definita, a è definita	

8. **4 punti** Scrivere cosa stampa il seguente programma.

```

1 int a= 3;
2 int f1(int* b, int c) {
3     static int a= 5;
4     if (a <= 6) {
5         int d= a++ + ++(*b) % c--;
6         printf("%d %d %d %d\n", a, *b, c, d);
7         d= f1(&c, *b);
8     } return d; }
9     else return a; }
10 int f2(int* b, int c, int* d) {
11     int e= a++ * ((*b)+=1) + ++(*d);
12     *d= f1(&c, a+1);
13     printf("%d\n", a);
14     return *d*2; }
15 int main(void) {
16     int a= 3, b= 2, c= 1;
17     b= f2(&a, b, &c);
18     printf("%d %d %d\n", a, b, c); }

```

6 3 4 8  
7 5 2 8  
4  
4 14 7

9. **5 punti** Cerchiare le affermazioni vere dato  $\text{int } a[5] = \{13+2*32, 129, [2] = \text{INT\_MIN} + 47, 130939, 131072*2+65\}$ ;  $\text{short int } *p = (\text{short}*) a$ ;  $\text{char } *q = (\text{char}*) a$ ;  $q[2] = -1$ ;  $p[3] = 128*2$ ; sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e  $131072 = 2^{17}$  (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Gli operatori  $|$  e  $\&$  ritornano rispettivamente l'or e l'and bit-a-bit dei due operandi,  $\sim$  è la negazione bit a bit, mentre  $>>$  ( $<<$ ) rappresenta l'operatore di *shift* di  $n$  posizioni a destra (sinistra), inserendo 0 nelle posizioni eliminate (operazione fatta nel processore).
- A.  $(\&a[4] - (a+1)) + *(q+18) - 7$  B.  $((\text{int})(a+7) - (\text{int})(p+3)) + q[16] \% 87$  C.  $(*(q+2) | q[8]) + (q[7] \& q[13])$   
D.  $((q[8] >> 3) + \sim q[4]) \% 130$  E.  $(q[13] - q[14] + *(p+7)) \% 2$  F.  $(*(p+8) - *(q+2) - q[6]) > 65$