Prova scritta Programmazione I - 21 Aprile **FILA A**

Nome e Cognome: _____ Matricola: _____

1. [2 punti] Elencare tutte le conversioni di tipo. Quanto vale *a* alla fine?

```
1 int a = 2.5;
2 short int b= 2L;
3 b= b + 1;
4 b += a;
5 a = 13 % b;
```

Causa Integer promotion

2.5 DA DOUBLE A INT
2L DA LONG A SHORT
b DA SHORT A INT, b+1 DA INT A SHORT
b DA SHORT A INT
a VALE 3

LONG
b DA SHORT A INT
a+b DA INT A SHORT

2. [3 punti] Scrivere cosa stampa il seguente programma.

```
1 int i= 6, a= 06;
2 while (a++,a-=2) {
3   a-=2;
4   i--;
5   printf("OK\n");
6   if (!i)
7   break;
8 }
9 int b= 0xff;
10 b+= a || a++;
11 printf("%d %d\n", a, b);
```

OK
OK
OK
OK
OK
OK

-12 256

3. [4 punti] Scrivere cosa stampa il seguente programma.

```
- * * - - * *
* * * * * * * *
* * * * * * *
- * * * * *
-- * * *
-- - *
```

```
1 int i, j, n = 4;
2 for(i=n/2; i<=n; i+=2) {
3   for(j=1; j<n-i; j+=2)
4     printf("-");
5
6   j= 1;
7   while(j<=i) {
8     printf("*");
9     j++;   }
10
11  for(j=1; j<=n-i; j++)
12    printf("-");
13
14  for(int j=1; j<=i; j++)
15    printf("*");
16
17  printf("\n");
18 }
19
20 for(i=n; i>=1; i--) {
21   for(j=i; j<n; j++)
22     printf("-");
23
24 for(int j=1; j<=(i*2)-1; j
       ++)
25   printf("*");
26
27 printf("\n");
28 }
```

4. [2 punti] Rappresentare per esteso in binario (dal primo all'ultimo byte) l'array multi-dimensionale *short int* a[2][2][3] = {{{1}, {4}}, {{7, 8}}}. Quanti byte occupa (uno *short* occupa 2 byte)?

	[0]	[1]	[2]
a[0][0]	1	0	0
a[0][1]	4	0	0

	[0]	[1]	[2]
a[1][0]	7	8	0
a[1][1]	0	0	0

OCCUPA 24 BYTE

5. [3 punti] Scrivere un un ciclo che inizializza una matrice *float mat[3][5]* come rappresentato in figura.

	[0]	[1]	[2]	[3]	[4]
mat[0]	0.0	0.1	0.2	0.3	0.4
mat[1]	1.0	1.1	1.2	1.3	1.4
mat[2]	2.0	2.1	2.2	2.3	2.4

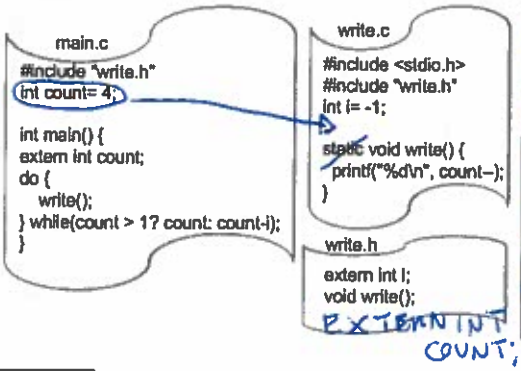
```
FOR (INT RIGA =0; RIGA <3; ++RIGA)
  FOR (INT COL=0; COL<5; ++COL)
    MAT[RIGA][COL]= RIGA + FLOAT(COL/10);
```

6. [4 punti] Scrivere cosa stampa il seguente programma.

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

2 3 1 6
3 4 0 48
4 5 -1 0
5 5 0

7. [4 punti] Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -o main main.c*, 2) *gcc -c main.c*, 3) *gcc -c write.c*, 4) *gcc -o main write.c main.c*. In caso il punto 4) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno *count*, *i*, e *write* in *main.c* e *write.c*, ed in quale file sono definite? Cosa stampa il programma?

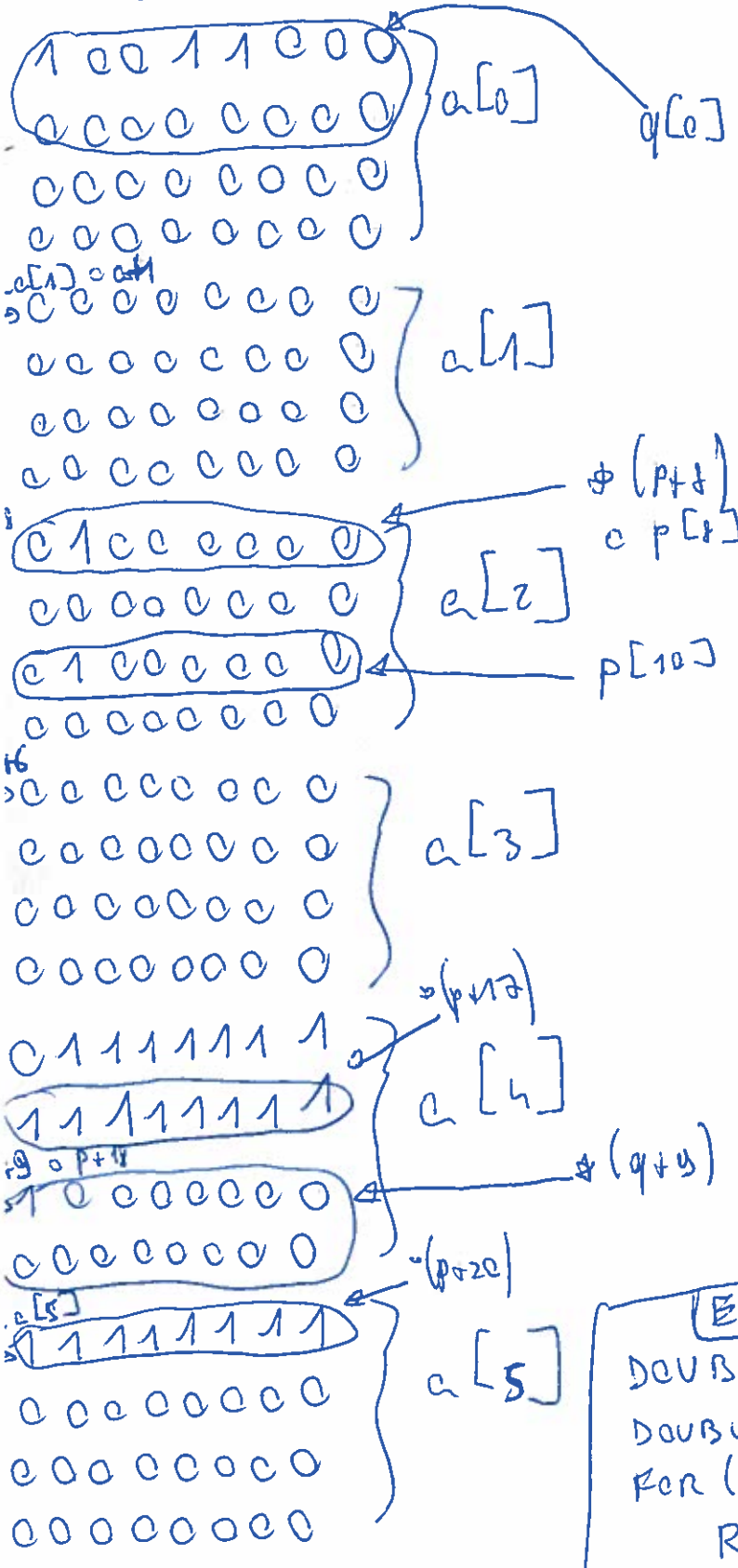


1) ERRORE MANCA DEF WRITE E i
2) OR 3) COUNT NON DEFINITE E WRITE PRIMA DICHIARATA COME EXTERN (IN WRITE.H) E POI DEFINITA COME STATIC (IN WRITE.C)
4) STESSI ERRORI DI 3) CORREGGENDO LA DEFINIZIONE DI COUNT IN WRITE.C E CANCELLANDO STATIC IN WRITE.C
STAMPA 4 3 2 1 0

8. [4 punti] Su foglio protocollo, scrivere una funzione di nome *a_average* che prende due parametri: un *int* array* ed un *int n*. Essi rappresentano un array e la sua lunghezza (numero elementi di tipo *int* contenuti nell'array); la funzione ritorna un *double*, che corrisponde alla media armomica dei valori x_i dell'array: $n \div \sum_{i=0}^{n} \frac{1}{x_i}$.

9. [4 punti] Cerchiare le affermazioni vere dato
INT *a[6]={25,[2]=131074,[4]= 131070, 255}; char *p = (char*) a; short int *q = (short*) a;*
sapendo che i tre tipi usati occupano 4, 1, e 2 byte, e 131072 = 2^{17} (valori rappresentati in *little endian*).
(A.) $*(p+8) + q[0] > 24$; B. $p[8] - p[10] > 0$; (C.) $((int)(q+6) - (int)(\&a[1]) - *(q+9))\%2 == 1$;
(D.) $\&a[5] - (a+1) + *(p+18) <= 5$; E. $*(p+20) - *(p+17) < 0$). F. $*(p+17) >= 0$.

ES 9



A VERA
- *(p+8) == 2
- q[0] = 25
- *(p+8) + q[0] = 27

B FALSA
- p[8] == 2
- p[10] == 2

p[8] - p[10] = 0

C VERA
SUPPONENDO CHE L'ARRAY È MEMORIZZATO DA INDIRIZZO DI MEMORIA 1000
- q+6 == 1012
- (int)(q+6) - (int)a[1] = 8
- a[1] == 1004
- *(q+9) == 1
- 8 - 1 == 7 % 2 == 1

D VERA
- a[5] - (a+1) == 4 È LA DIFFERENZA IN NUMERO DI ELEMENTI NELL'ARRAY
- *(p+18) == 1
- 4+1 == 5 <= 5

E FALSA
- *(p+20) == *(p+18)

LA LORO DIFFERENZA È 0

F FALSA
- *(p+17)

BIT PIÙ SIGNIFICATIVO A 1 -> NUMERO NEGATIVO

ES 8

```
DOUBLE A_ARMONICA (INT* ARRAY, INT N){
DOUBLE RES = 0.0
FOR (INT i = 0; i<N; i++)
    RES += (DOUBLE) 1/ARRAY[i]

RES = N / RES ;
RETURN (RES);
}
```

Prova scritta Programmazione I - 21 Aprile **FILA B**

Nome e Cognome: _____ Matricola: _____

1. [2 punti] Elencare tutte le conversioni di tipo. Quanto vale a alla fine?

```
1 int a = 2;
2 short int b= 2.5;
3 b= b + 1;
4 b += a;
5 a = 16 % b;
```

LINEA 2

VEDI FILA A b DA DOUBLE A SHORT INT

a VALE 1

2. [3 punti] Scrivere cosa stampa il seguente programma.

```
1 int i= 5, a= 13;
2 while (a++,a-=2) {
3   a-=2;
4   i--;
5   printf("OK\n");
6   if (!i)
7     continue;
8 }
9 int b= 0xff;
10 b+= ++a || a++;
11 printf("%d %d\n", a, b);
```

OK
OK
OK
OK
1 256

3. [4 punti] Scrivere cosa stampa il seguente programma.

```
1 int i, j;
2 char alph = 'A';
3 int n= 5, blk;
4 int ctr = 1;
5
6 for (i = 1; i <= n; i++) {
7   for(blk=1;blk<=n-i;blk++)
8     printf("--");
9
10  for (j=0; j<=(ctr/2); j++)
11    printf("%c-", alph++);
12
13  alph = alph - 2;
14
15  for (j=0; j<(ctr / 2); j
      ++) {
16    printf("%c-", alph--);
17  }
18
19  ctr = ctr + 2;
20  alph = 'A';
21  printf("\n");
22 }
23
```

--------A-
------A-B-A-
----A-B-C-B-A-
--A-B-C-D-C-B-A-
A-B-C D-E-D-C-B-A

4. [2 punti] Rappresentare per esteso in binario (dal primo all'ultimo byte) l'array multi-dimensionale *short int* $a[2][2][3] = \{1, [0][1][0] = 4, [1][0][0] = 7, 8\}$. Quanti byte occupa (uno *short* occupa 2 byte)?

VEDI FILA A

5. [3 punti] Scrivere un un ciclo che inizializza una matrice *float mat[3][5]* come rappresentato in figura.

	[0]	[1]	[2]	[3]	[4]
mat[0]	0.0	0.1	0.2	0.3	0.4
mat[1]	1.0	1.1	1.2	1.3	1.4
mat[2]	2.0	2.1	2.2	2.3	2.4

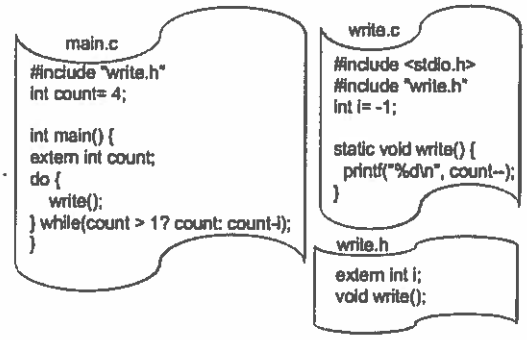
VEDI FILA A

6. [4 punti] Scrivere cosa stampa il seguente programma.

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

2 3 1 6
3 4 0 0
4 5 -1 0
0 0 0

7. [4 punti] Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -o main main.c*, 2) *gcc -c main.c*, 3) *gcc -c write.c*, 4) *gcc -o main write.c main.c*. In caso il punto 4) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno *count*, *i*, e *write* in *main.c* e *write.c*, ed in quale file sono definite? Cosa stampa il programma?



VEDI FILA A

8. [4 punti] Su foglio protocollo, scrivere una funzione di nome *average* che prende due parametri: un *int* array* ed un *int n*. Essi rappresentano un array e la sua lunghezza (numero elementi di tipo *int* contenuti nell'array); la funzione ritorna un *double*, e cioè la media dei valori x_i dell'array: $(\sum_{i=0}^{n} x_i) \div n$.

9. [4 punti] Cerchiare le affermazioni vere dato
*a[6]= 25,[2]=131074,[4]= 131070, 255; char *p = (char*) a; short int *q = (short*) a;*
sapendo che i tre tipi usati occupano 4, 1, e 2 byte, e $131072 = 2^{17}$ (valori rappresentati in *little endian*).
A. $*(p+17) >= 0$. (B.) $*(p+8)+q[0] > 24$; (C.) $\&a[5]-(a+1)+*(p+18) <= 5$; (D) $((int)(q+6)-(int)(\&a[1])-*(q+9))\%2 == 1$; E. $*(p+20)-*(p+17) < 0$). F. $p[8]-p[10] > 0$;

VEDI FILA A : PUNTI INVERTITI