Prova scritta Programmazione I - 20 Giugno

Nome e Cognome: _____ Matricola: _____

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

Linea 4: p2 da short a int per integer promotion

```
1 #define A 2.5
2
3 short int f(double p1, short int p2){
4   return (p1 >= p2 ? p1 : p2);
5 }
6
7 int main(void) {
8   float a = A;
9   int b= 2LL;
10  a= f(a, b);
11 }
```

LINEA 8: 2,5 DA DOUBLE A FLOAT
LINEA 9: 2 DA [redacted] INT A INT
LINEA 10: A DA FLOAT A DOUBLE E B DA INT A SHORT INT IN CHIAMATA F(A,B)
LINEA 4: RITORNO F DA DOUBLE A SHORT
LINEA 4: P2 DA INT A DOUBLE
A VALE 2.0

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

```
1 int i, a;
2 for (i= 6, a= 011 ; i--, i >=0; printf("OK %d %d\n", i, a)) {
3   a-= a / 2;
4 }
5 int b= 0xfa;
6 b+= a;
7 printf("%d %d\n", a, b);
```

ok 5 5
ok 4 3
ok 3 2
ok 2 1
ok 1 1
ok 0 1
1 251

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

```
1 int r= 5, c= 7, i, j;
2
3 for(i = 0; i < r; i++){
4   for(j = 0; j < c; j++){
5     if(i==0 || i==r-1 || j==0 || j==c-1) {
6       printf("0");
7       continue;
8     }
9     else
10      printf("1");
11  }
12 printf("\n");
13 }
```

0 0 0 0 0 0 0
0 1 1 1 1 1 0
0 1 1 1 1 1 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0

4. [2 punti] Cerchiare tra i seguenti punti, quelli che rappresentano un *rvalue*, dato *int* $a[5]$; *int* $*p = a$; A. $a[1]$ B. $a[0]+1$ C. p D. $*p$ E. $p+1$ F. $*p+1$ G. 7 H. $(\&a[4]) - a$

5. [3 punti] Cosa stampa il programma? a si trova all'indirizzo di memoria *0x7fff5eb9bff5*, un *long long* occupa 8 byte, un *int* 4 byte.

```
1 long long int a = -3, *b= &a;
2 int c = (a+=1+1, a + 1,  a ? (a++, 1) : 2);
3 printf( "%lld %d \n", a, c);
4  printf("%p %p %lu\n", b, (int *) b + 3,
     sizeof(*b));
5
```

0 1
0x7fff5eb9bff5 0x7fff5eb9bff8

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

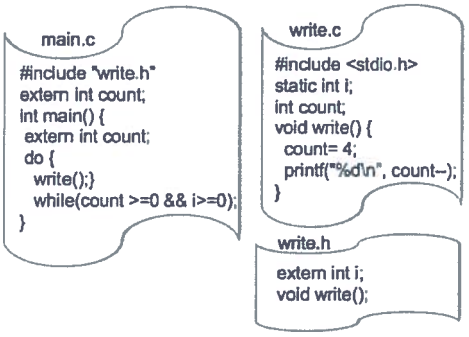
```
1  int a= 3;
2
3  int f1(int* b, int c) {
4    int a= 1;
5    int d= ++a + (*b)-- - ++c ;
6    printf("%d %d %d %d\n", a, *b, c, d);
7    return d;
8  }
9  int f2(int* b, int c, int* d) {
10   int e= ++a * (*b)++  * c++ * (*d)++;
11   *d= f1(&c, a);
12   printf("%d\n", a);
13   return e + *d;
14 }
15 int main(void) {
16  int a= 2, b= 3, c= 1;
17  b= f2(&a, c, &b);
18  printf("%d %d %d\n", a, b, c);
19 }
```

2 1 5 -1

4

3 23 1

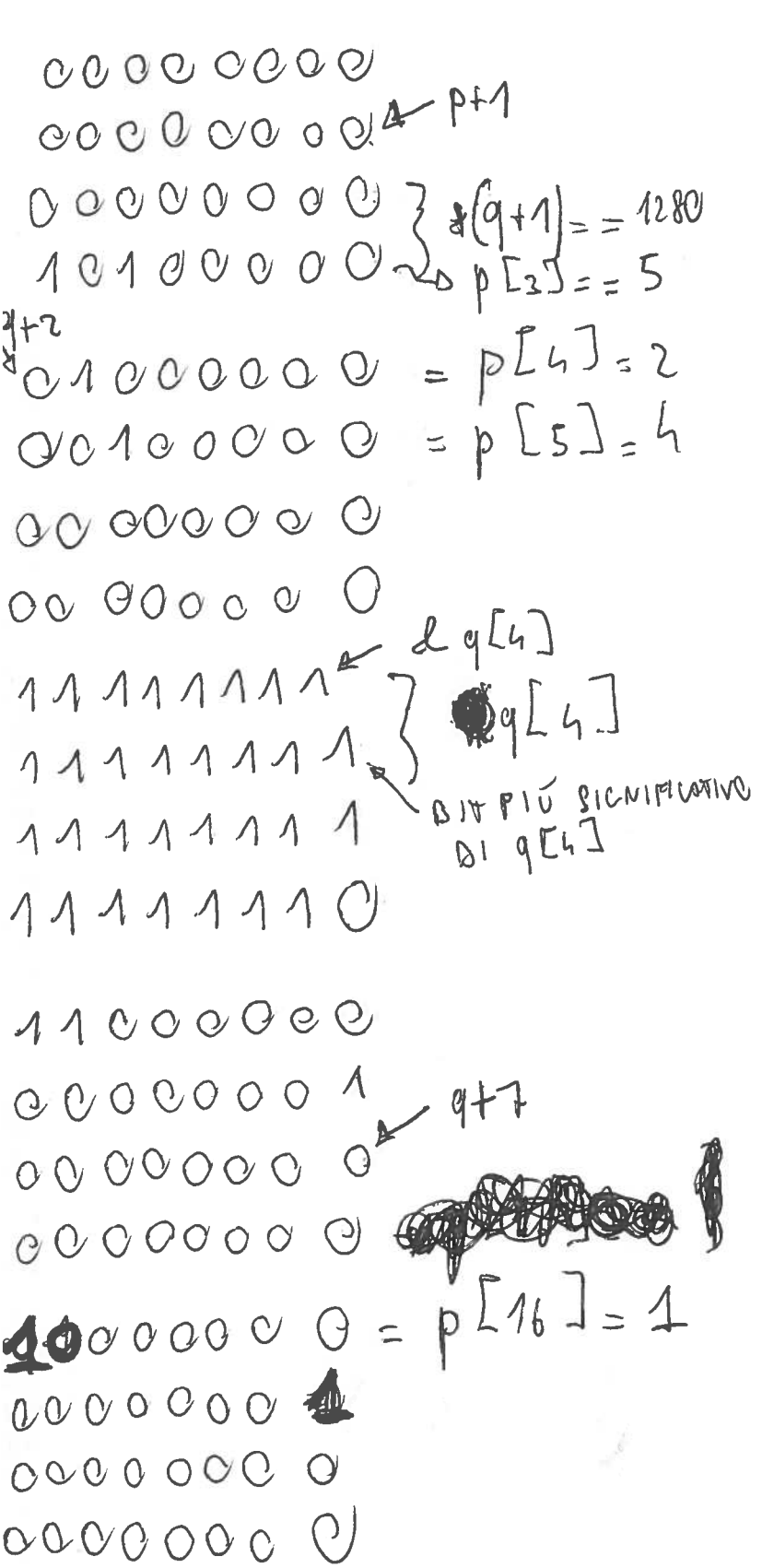
7. [4 punti] Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc -o main main.c*, 2) *gcc -c main.c*, 3) *gcc -o main write.c*, 4) *gcc -o main write.c main.c*. In caso il punto 4) ritorni un errore, descrivere come può essere corretto eliminando una sola *keyword*. Dopo aver 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 DEFINIZIONE COUNT, i, WRITE
2) OK 3) ERRORE MANCA DEFINIZIONE MAIN
4) ERRORE i È STATIC QUINDI NON VISIBILE IN MAIN.C
PER RISOLVERE, CANCELLARE STATIC DA STATIC INT i
i, WRITE E COUNT HANNO LINKAGE ESTERNO
E SONO DEFINITE IN WRITE.C, STAMPA 4 INFINITE VOLTE

8. [4 punti] Su foglio protocollo, scrivere una funzione di nome *fillMatrix* che prende tre parametri: *int* m*, *int rows*, e *int cols*, che rappresentano rispettivamente il puntatore alla matrice, il suo numero di righe, e quello delle colonne. La funzione assegna ad ogni cella della matrice la somma del suo indice di riga più l'indice della sua colonna, se questa somma è minore o uguale a 5; altrimenti assegna il reciproco di tale somma alla cella. Esempio: il contenuto di *m[1][3]* deve essere uguale a 4, quello di *m[3][3]* a $1 \div 6$.

9. [4 punti] Cerchiare le affermazioni vere dato
*int a[5]= {[1]= 1026, INT_MAX, 32771, [4]= 32769}; char *p = (char*) a; short int *q = (short*) a; p[3]= 5;*
sapendo che i tre tipi usati occupano 4, 1, e 2 byte, e 32768 $= 2^{15}$ (valori rappresentati in *little endian*).
A. $p[4]-p[16]! = 1$; B. $*(q+1)+p[4] < 7$; C. $((int)(q+7)-(int)(\&*(p+1)))\%2$; D. $p[3]-p[5]-p[16] > 0$;
E. $(q+2) - \&q[4] >= 0$. F. $q[4] >= 0$.



A. p[4] - p[16] == 1

B. *(q+1) + p[4] == 1282

C. 13 % 2 == 1

D. 5 - 4 - 1 == 0

E. & q[4] È MEMORIZZATO AD UN INDIRIZZO SUCCESSIVO RISPETTO A q+2: LA DIFFERENZA È NEGATIVA

F. q[4] È UN VALORE NEGATIVO: IL BIT PIÙ SIGNIFICATIVO È 1

```c
#include <stdio.h>


void fillMarix(int rows, int cols, double m[rows][cols])
{

    for (int i= 0; i < rows; i++)
        for (int j= 0; j < cols; j++)
            if (i+j <= 5)
                m[i][j]= i+j;
            else
                m[i][j]= (double) i / j;
}


int main() {
    int n_rows= 0;
    int n_cols= 0;

    scanf("%d", &n_rows);
    scanf("%d", &n_cols);
    double m[n_rows][n_cols];
    fillMarix(n_rows, n_cols, m);
    for (int i= 0; i < n_rows; i++) {
        for (int j= 0; j < n_cols; j++)
            printf("%f ", m[i][j]);
        printf("\n");
    }
}
```