Prova scritta Programmazione I FILA A - 9 Febbraio 2018.

Nome e Cognome: _____ Matricola: _____

1. [3 punti] Riportare le conversioni di tipo -implicite- e scrivere cosa stampa il programma.

```
1 int x = 0, i = -2;
2 char a= (char) 80, b= (char) 80, c= (
      char) 100;
3 a= (a*b) / c;
4 unsigned int limit = 10U;
5 long n = 20L;
6 if ( i < limit )
7   x = limit * n;
8 printf("%d %d\n", a, i);
```

LINEA 3: a, b c CONVERTITI DA CHAR A INT PER INTEGER PROMETION. RISULTATO CONVERTITO A char
LINEA 6: i CONVERTITO DA INT A UNSIGNED INT
LINEA 7: LIMIT E N CONVERTITI ENTRAMBI A UNSIGNED LONG (SE INT E LONG 4 BYTE), ALTRIMENTI LIMIT CONVERTITO A LONG
STAMPA 64 -2

Linea 7: Risultato convertito a int (x)

linea 7: risultato da unsigned long a int

ESERCITO IDENTICO A SLIDE SU CONVERSIONE DEI TIPI

2. [4 punti] Scrivere cosa stampa il seguente programma.

```
1  int i, j, N= 5;
2    for(i=1; i<=N; i++) {
3      for(j=1; j<i; j++)
4        printf("-");
5        printf("%d", i);
6      for(j=1; j<=((N - i) * 2
       - 1); j++)
7        printf("-");
8      if(i != N)
9        printf("%d", i);
10     printf("\n");;
11   }
12   for(i=N-1; i>=1; i--) {
13     for(j=1; j<i; j++)
14       printf("-");
15     printf("%d", i);
16
17     for(j=1; j<=((N - i ) * 2
       - 1); j++)
18       printf("-");
19       printf("%d\n", i);
20   }
```

```
1-------1
-2-----2
--3---3
---4-4
----5
---4-4
--3---3
-2-----2
1-------1
```

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

```
1 int a= 0x35, *b= &a, i= 00;
2 for (int i= 010; (a-= 1) ? (i++, --a) : (i+=2, a--, a--); i++) {
3   a= (a - i);
4   printf("%d OK\n", a);
5   if (a<0) {
6     a= -*b + i;
7     break; }
8 }
9 a= -a || i++;
10 printf("%d %d\n", a, i);
```

42 OK
29 OK
14 OK
-3 OK
1 0

4. [4 punti] Dato il seguente *main*, definire la funzione *matrix_vect_multiply* che calcola in *v2* il risultato della moltiplicazione di *v1* per la matrice *m*. Nella funzione, controllare per prima cosa se la lunghezza di *v1* (cioè il parametro (*2*) prima di *v1*) è diverso dal numero di righe (se sì, stampare un messaggio ed uscire subito dalla funzione). I parametri *2* e *3* prima di *m* corrispondono rispettivamente al numero di righe e colonne. Il seguente programma alla fine stamperà quindi 11, 16, 19. *Sugg: v1 va moltiplicato per ogni colonna di m.*

```
1 int main() {
2   int m[2][3]= {1,4,5,2,1,1};
3   int v1[2]= {3, 4};
4   int v2[3]= {0, 0, 0};
5   matrix_vect_multiply(2, 3, m, 2, v1, v2);
6   for (int i= 0; i< 3; i++)
7     printf("%d ", v2[i]);
8 }
```

```
VOID MATRIX_VECT_MULTIPLY (INT ROWS, INT COLUMN, INT M[ROWS]
    [COLUMN], INT LUN, INT V1[2], INT V2[2]) {
IF (LUN != ROWS) {
    PRINTF ("ERRORE!\n");
    RETURN;
}
ELSE
    FOR(INT J=0; J<COLUMN; J++)
        FOR(INT I=0; I<ROW; I++)
            V2[J] += M[I][J] * V[I];
}
```

5. [3 punti] Elencare tutti gli effetti collaterali su ogni variabile, ed evidenziare i *sequence point* con una freccia.

```
1 int a, b, *p= &b;
2 a= a && b++;
3 p= &a;
4 b= ((*p++) ? (b++, b--) : (a= 3));
```

LINEA 2: UN EFFETTO SU A (=) UNO SU B (++)
LINEA 3: UN EFFETTO SU P (=)
LINEA 4: UN EFFETTO SU A (*P++), TRE SU B (++ E --)
SE LA CONDIZIONE È VERA, UNO SU A (=) SE FALSA

6. [3 punti] Scrivere una funzione ricorsiva di nome *GCD* che ha come parametri due interi *n1* e *n2* e restituisce il massimo comune divisore dei due numeri. *Sugg: seguire il procedimento dell'esempio in figura.*

GCD(206,40) = GCD(40,6)
= GCD(6,4)
= GCD(4,2)
= GCD(2,0)
= 2

```
INT GCD (INT X, INT Y) {
  IF (Y==0)
    RETURN (X);
  ELSE {
    INT RES = GCD (Y, X%Y);
    RETURN (RES);
  }
}
```

7. [4 punti] Scrivere cosa stampa il seguente programma.

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

5 16 3 18
5 12 3 30
5 8 3 37
5 37 3

8. [4 punti] Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc –c write.c*, 2) *gcc –o main main.c*, 3) *gcc –o write write.c*, 4) *gcc –c main.c*, 5) *gcc write.c main.c -o main.* In caso il punto 5) ritorni un errore, descrivere come può essere corretto. Dopo aver corretto l'errore, che tipo di *linkage* hanno *count*, *i*, e ~~*write*~~, ed in quale file sono definite? Cosa stampa il programma?

MY_WRITE

main.c
```
int count= 4;
void mywrite(int count);

int main(void) {
  int count= 4;
  do {
    mywrite(count);
  }while(count >=0);
}
```

write.c
```
#include <stdio.h>
extern int count;

void mywrite(int a) {
  static int i= 1;
  printf("%d\n", count=
                 count - (a + i));
}
```

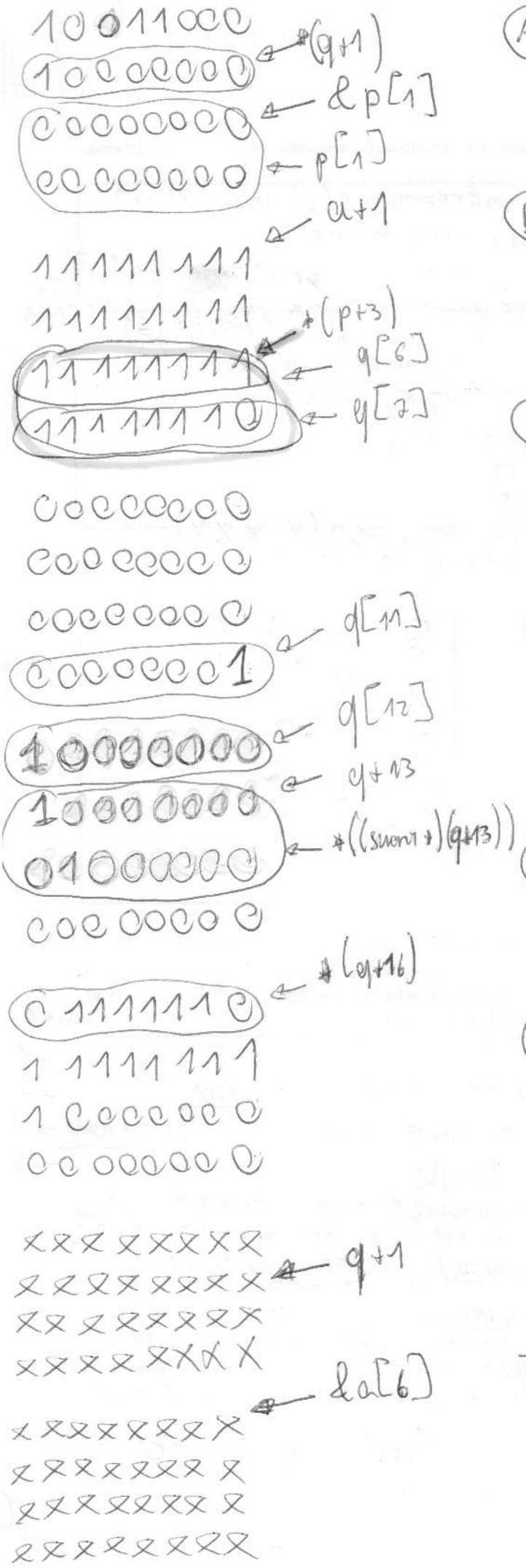
1) OK
2) MY_WRITE NON DEFINITA ERRORE
3) FUNZIONE MAIN E COUNT NON DEFINITI ERRORE
4) OK 5) OK
COUNT HA LINKAGGIO ESTERNO IN MAIN.C E WRITE.C
MY_WRITE HA LINKAGGIO ESTERNO IN MAIN.C E WRITE.C
I HA NO LINKAGE / COUNT DEFINITO IN MAIN.C

STAMPA
-1
-6
-11
-16

MY_WRITE DEFINITA IN WRITE.C

9. [4 punti] Cerchiare le affermazioni vere dato *int a[5]= {281, INT_MAX, [2]= INT_MIN, 131329, 130942}; short int *p = (short*) a; char *q= (char*) a;* sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e 131072 = 2^{17} (valori rappresentati in *little endian*). (A.) $(*(q+1)-q[12]+*(p+3))\%2$ B. $(q[11]+*(q+7)-q[6])>0$ (C.) $(\&a[6]-(a+1))\%2$ D. $(int)(q+21)-(int)(\&(p[1]))-q[1]>18$ E. $p[1]\ ?\ q[1]\ :\ *(q+16)>127$ (F.) $*((short^*)(q+13))>500$

I DEFINITA IN WRITE.C



(A) $1 - 1 + {}^{A}(p+3) \,\%\, 2 == 1$

DISPARI: ULTIMO BIT!

VERA

(B) $q[11] + *(q+7) == q[6]$

QUINDI

$q[11] + *(q+7) - q[6] == 0$

FALSA

(C) TRA a+1 E &a[6] CI SONO 5 INTERI. $5 \,\%\, 1 == 1$

VERA

(D) $(q+1) - \&p[1] == 19$

$19 - q[1] == 18$

$= 1$

FALSA

(E) $p[1] == 0$ (FALSO) QUINDI

$*(q+16) > 128$

$= 126$

FALSA

(F) SE CONVERTITO A PUNTATORE A SHORT, $*((short *)(q+13))$ CORRISPONDE A PRENDERE IL VALORE DEI DUE BYTE A PARTIRE DA INDIRIZZO $q+13$. $513 > 500$

VERA

IL VETTORE a CORRISPONDE AI PRIMI 20 BYTE, I SUCCESSIVI BYTE SONO SCONOSCIUTI (X)