

Prova scritta Programmazione I FILA A - 18 Gennaio 2018.

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

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

1. **3 punti** Aggiungere al seguente programma operatori di conversione esplicite (*cast*) ovunque ci sia una conversione implicita. Dati `UINT_MAX = 4294967295` e `USHRT_MAX = 65535`, scrivere i valori di `a` e `d`.

```
1 double pow2(float a) {
2   return (a * a);
3 }
4 int main(void) {
5   unsigned a = -1;
6   float b = pow2(a);
7   double *p = malloc(sizeof(double));
8   unsigned short d = -4;
9 }
```

`unsigned a = (unsigned int) -1;`  
`float b = (float) pow2((float) a);`  
`double *p = (double *) malloc(sizeof(double));`  
`unsigned short d = (unsigned short) -4;`  
`((double) (a)) * ((double) (a))`  
`a == 4294967295 (UINT_MAX)`  
`d == 65532 (USHRT_MAX-3)`

2. **4 punti** Scrivere cosa stampa il seguente programma.

```
1 int i, j, s, r=8, st=0;      11 printf("\n");
2 for(i = 1; i <= r; i++) {  12 } else {
3   if(i <= 5) {              13   for(j = 0; j < 9; j++) {
4     for(s = 1; s <= 5-i; s++) 14     if((int)(j/3) == 1)
5       printf("-");           15     printf("-");
6     while(st != (2*i - 1))    16     else
7     {                         17     printf("*");
8       printf(" ");           18   }
9     st++;                     19   printf("\n");
10  }                           20 }
11  st=0;                       21 }
```

```

- - - - *
- - - * * *
- - * * * *
- * * * * *
* * * * *
* * * - - *
* * - - - *
* - - - - *
- - - - -

```

3. **3 punti** Scrivere cosa stampa il seguente programma.

```
1 int a= 0x8;
2 while(a > 8 ? a++: (a--, --a)) {
3   printf("OK\n");
4   if (a + 2 >= 07) {
5     continue;
6     printf("NO MORE OK\n");
7   }
8 }
9 a = a++ && a++;
10 a+= 0xae;
11 printf("%d\n", a);
```

OK  
 OK  
 OK  
 174

4. **3 punti** Dato il seguente *main*, definire la funzione *matrix\_avg\_col* che calcola in *avg\_col* la media dei valori della matrice per colonna: il seguente programma alla fine stamperà quindi 1.5, 2.5, 3.0.

```
1 int main() {
2   int m[2][3] = {1,4,5,2,1,1};
3   double avg_col[3];
4   matrix_avg_col(2, 3, m, avg_col);
5   printf("%f\n", avg_col[0]);
6   printf("%f\n", avg_col[1]);
7   printf("%f\n", avg_col[2]);
8 }
9
```

`void matrix_avg_col(int row, int column, int m[][column], double avg_col[])`  
`for (int j = 0; j < column; j++)`  
`for (int i = 0; i < row; i++)`  
`avg_col[j] += m[i][j];`  
`for (int i = 0; i < column; i++)`  
`avg_col[i] /= row;`  
`}`

5. **3 punti** Scrivere una singola espressione con 3 effetti collaterali su una variabile  $a$  e 2 effetti collaterali su  $b$ , che *NON* generi un warning *multiple unsequenced modifications*, evidenziando i *sequence points* con una freccia. Scrivere una seconda espressione con 2 effetti collaterali su  $a$  che invece generi tale warning.

$b = a++$ ,  $a++$ ,  $a--$ ,  $b++$  (1)  
 $c = a++$  (2)

6. **3 punti** Su foglio protocollo, scrivere una funzione ricorsiva di nome *fact* che ha come parametro un intero  $n$  e ne restituisce il fattoriale. Qual è il numero massimo di copie (*frame*) della funzione aperte nello stack con  $n == 5$ ? Quanti differenti  $n$  vengono allocati nello stack?

```

INT FACT (INT N) {
    IF (N <= 1)
        RETURN 1;
    ELSE
        RETURN (FACT(N-1) * N);
}

```

7. 4 punti Scrivere cosa stampa il seguente programma.

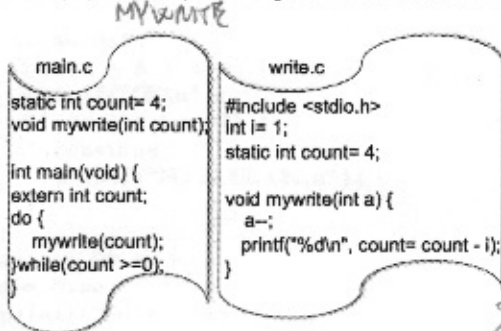
```

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

```

5	6	8	2
5	6	24	4
5	6	38	2
2	44	6	

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?



1) OK 2) ERRORE MANCA DEFINITIVE DI MYWHITE  
3) ERRORE MANCA DEFINITIVE MAIN 4) OK  
5) OK NON C'E ERRORE IN 5; ESISTONO DUE  
COUNT DIFFERENTI, UNO PRIVATE IN MAIN.C E  
UNO PRIVATE IN WHITE.C (INTERNAL LINKAGE)  
2 COUNT DEFINITI IN MAIN.C E WHITE.C; I DEFINITI IN WHITE  
MYWHITE DEFINITO IN WHITE.C MYWHITE EXTERNAL LINK  
i external linkage

9. **4 punti** Cerchiare le affermazioni vere dato `int a[5] = {25, [2] = INT_MAX, 131046, 131328};` *IL PROBLEMA VA IN*  
`short int *p = (short*) a;` `char *q = (char*) a;` sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e `131072 =` *LOE P*  
`217` (valori rappresentati in *little endian*). **A.** `(*p - q[17] - *(q + 18)) % 7` **B.** `(*p + 4) + *(q + 14) + q[11] ==`  
`128` **C.** `((&a[4] - a) % 2) == 0` **D.** `(int)(a + 5) - (int)(p + 6) + q[5] == 2` **E.** `q[8] - q[9] + q[10] + q[11] >= 126` **3**  
**F.** `p[2] || *(q + 10)` **2**  
**1**  
**0**  
**-1**

☐ step; ☐ nocall; ☒ next; ☐ walkthrough; ☐ nojump; ☐ command.

100 11000	00000000	1111111	01100111	00000000
000 00000	000 00000	11111111	11111111	10000000
00000000	00000000	11111111	10000000	01000000
00000000	00000000	11111110	00000000	00000000

ES 9

$a = p = q$

$\rightarrow p$

10011000  
00000000  
00000000  
00000000

$\rightarrow p[2]$

00000000  
00000000  
00000000  
00000000

$q[5]$

11111111

$q[8]$

11111111

$q[9]$

11111111

$q[10] \rightarrow (q+10)$

11111111

$q[11]$

$p+6$

01100111  
11111111  
10000000  
00000000

$\rightarrow (q+14)$

$q[4]$

00000000  
10000000  
01000000  
00000000

$q[17]$

$\rightarrow (q+18)$

$a+5$

-----  
-----  
-----  
-----

A  $(25-1-2) \% 7 = 1$  VERO

B  $-1 + 1 + 127 = 128$  FALSE

C  $4 \% 2 = 0$  VERO

D  $8 + 0 = 2$  FALSE

E  $-1 - (-1) + (-1) + 127 = 126$  VERO

F  $0 \parallel -1$  VERO