

Prova scritta Programmazione I - 24 Giugno 2019.

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

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

1. **3 punti** Riportare le conversioni di tipo implicite. Dati  $UINT\_MAX = 4294967295$  e  $USHRT\_MAX = 65535$ , scrivere i valori di  $a$  e  $d$ .

```
1 double pow2(float a) {
2     return (3 * a); }
3
4 int main(void) {
5     unsigned a = -2;
6     long double b= pow2(a);
7     double *p= malloc(sizeof(float));
8     unsigned short d = -11;
9 }
```

linea 5: -2 da int a unsigned int  
 linea 6: a da unsigned int a float, risultato funzione da double a long double  
 linea 7: risultato malloc da void\* a double\*  
 linea 8: -11 da int a unsigned short  
 linea 2: 3 da int a float, 3\*a da float a double

**a = 4294967294**  
**b = 65525**

2. **3 punti** Scrivere cosa stampa il seguente programma.

```
1 int i, j, s, r=6, st=0;          11 printf("\n");
2 for(i = 1; i <= r; i++) {      12 } else {
3     if(i <= 4){                  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("o");
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 }
```

```
----*
---***
--*****
-*****
***ooo***
***ooo***
```

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

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

```
HERE 7
HERE 5
EXIT
EXIT
a: 174
```

4. **3 punti** Dato il seguente *main*, su foglio protocollo definire la funzione *matrix\_avg\_diag* che calcola in *avg\_diag* la media dei valori della matrice *sulle due diagonali*: il seguente programma alla fine stamperà quindi 2 e 3. All'inizio della funzione, controllare se i parametri passati sono corretti per la logica della funzione.

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

utilizzare due cicli for indicizzati con i, dove n rappresenta la dimensione della matrice.  
 Se 5x5, n == 5  
 sum1 = m[i][i]  
 sum2 = m[i][n-i-1]

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

$a++, a--, b++, b++, b=3$

$a= a++, a++$

6. **3 punti** Su foglio protocollo, scrivere una funzione ricorsiva di nome  $fibN(b, n)$  che ha come parametro due interi  $b$  e  $n$ , e restituisce il numero di fibonacci a partire dalla base di partenza  $b$ .  $fibN(b, 0) = b$  e  $fibN(b, 1) = b+1$  Per esempio  $fibN(3, 0) = 3$ ,  $fibN(3, 1) = 4$ ,  $fibN(3, 2) = 7$ ,  $fibN(3, 3) = 11$ , etc.
7. **4 punti** Scrivere cosa stampa il seguente programma.

```

1 int f1(int* b, int c, int* e) {
2     static int a= 2;
3     if (a < 4) {
4         int res= (a++ + (*b)++ + (c+= *e + f1(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= f1(&c, b, &a);
14     printf("%d %d %d\n", a, b, c);
15 }

```

4 5 10 4  
4 5 24 2  
2 30 5

8. **4 punti** Per ogni identificatore di variabile e funzione scrivere se è definito o dichiarato, ed il suo linkage.

```

1 #define A 3
2 int a= A;
3 int a;
4 extern int b;
5 extern int cmp(float, float);
6 static double area(double);
7
8 int* my_func(int c) {
9     static double e= 4.0;
10    double* f= &e;
11    register int q= 4;
12    extern int a;
13    // Altri comandi
14 }

```

**a** linea 3 definita/ext  
**a** linea 3 tentativo di def, dichiarata/ext  
**b** dichiarata/ext  
**cmp** dichiarata/ext  
**area** dichiarata/int  
**my\_func** definita/ext  
**c** definita/nolinkage  
**e** definita/nolinkage  
**f** definita/nolinkage  
**q** definita/nolinkage  
**a** linea 12 dichiarata/~~nolinkage~~ **external**

9. **4 punti** Cerchiare le affermazioni vere dato  $\text{int } a[5] = \{25, [2] = \text{INT\_MAX}, 131046, 131328\};$   
 $\text{short int } *p = (\text{short} *) a;$   $\text{char } *q = (\text{char} *) a;$  sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e  $131072 = 2^{17}$  (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.**  $(\text{int})(a + 5) - (\text{int})(p + 6) + q[5] == 2$  **(E.)**  $q[8] - q[9] + q[10] + q[11] >= 126$   
**(F.)**  $p[2] || *(q + 10)$  **vedere correzione 18 gennaio 2018 (FILA A)**
10. **1 punti** Quali affermazioni sono vere? ☐ la keyword *static* è solamente legata al linkage; ☐ le espressioni contengono comandi; ☐  $(++a++)$  è un'espressione corretta. **nessuna è vera**

## Esercizio 6

```
int fibN(int b, int n)
{
    if (n < 0) return -1; /* fibN(i) non e' definito per interi i negativi! */

    if (n == 0) return b;
    else if (n == 1) return b+1;
    else return fibonacc(i-1) + fibonacc(i-2);
}
```