Prova scritta Programmazione I FILA A - 11 Giugno 2018.

Nome e Cognome: _____ Matricola: _____

1. [3 punti] Riportare tutte le conversioni di tipo implicite. Dare il valore di finale di d dato $UINT\_MAX = 4294967295$. In caso non sia possibile stabilire il valore di d, spiegare perché.

UINT_MAX + C - 7 HA PIÙ DI 6 CIFRE (CORRISPONDENTI ALLA PRECISIONE DI UN FLOAT; NON PUÒ ESSERE RAPPRESENTATO CON PRECISIONE

```
1 double f(float a) {return (a - 1);}
2
3 int main(void) {
4   unsigned  a =  3LL;
5   int b=  -1U;
6   float c= f((b < a)? b : a);
7   float d= UINT_MAX + c- 7;
8 }
```

LINEA 4: 3LL DA LONG LONG INT A UNSIGNED INT
LINEA 5: -1U DA UNSIGNED INT A INT
LINEA 6: b < a CONVERTITO DA INT A UNSIGNED
a DA UNSIGNED A FLOAT PASSATO COME PARAMETRO AF
LINEA 1: -1 DA INT A FLOAT, RISULTATO F DA FLOAT A DOUBLE
LINEA 6 RISULTATO F DA DOUBLE A FLOAT
LINEA 7: UINT_MAX DA UNSIGNED A FLOAT, 7 DA INT A FLOAT.

2. [3 punti] Scrive una funzione *main()* che disegna su terminale il seguente triangolo di lettere. Il numero di righe è preso come input da tastiera. *Suggerimento: utilizzare due cicli innestati.*

```
A
ABA
ABCBA
ABCDCBA
ABCDEDCBA
```

3. [3 punti] Scrivere cosa stampa il seguente programma, sapendo che a si trova all'indirizzo 0x7ffee9289918.

```
1   int a= 0xaf, i= -13, *b= &a;
2
3   for (int* p= &i; (a-= 3, a--, (*p)++) ? (++(*p), (a--)-2) : ((*
      p)+=3, a); (*p)++) {
4   --a;
5   printf("%d %d OK\n", a, *p);
6   if (*p > 0) {
7   a= (!(-a) || a++) ? 5 : 3;
8   break; } }
9   printf("%d %p %d\n", a, ((char*) b) + a, i);
10
```

169 -11 OK
163 -8 OK
157 -5 ok
151 -2 OK
145 1 OK

5 0x7ffee928899 1

4. [4 punti] Su foglio protocollo, Scrivere una funzione *main()* e una funzione *half_array()*. La funzione *main()* legge da tastiera un array di n posizioni (con n letto da tastiera) ciascuna di tipo *int*. L'array viene passato a *half_array()* che crea in memoria dinamica un nuovo array di *int* di lunghezza $n \div 2$. Il nuovo array viene riempito sommando aritmeticamente il primo elemento della prima metà dell'array passato, con il primo elemento della seconda metà dell'array passato (la somma è memorizzata in prima posizione del nuovo array); il secondo elemento della prima metà con il secondo elemento della seconda metà, e così via. Il nuovo array viene ritornato a *main()*, che lo stampa su schermo. In caso n sia dispari, non considerare l'ultimo elemento.

5. [3 punti] Su foglio protocollo, descrivere (anche con brevi esempi di codice) le differenze tra un linguaggio fortemente e debolmente tipato, e staticamente e dinamicamente tipato. SLIDE 7_TYPE_CHECKING

6. [3 punti] Su foglio protocollo, scrivere un solo file in cui siano presenti esattamente *i)* un tentativo di definizione che rimane dichiarazione, *ii)* un identificatore con *internal linkage*, *iii)* tre identificatori con *external linkage* (di cui un identificatore di funzione), *iv)* quattro identificatori con *no linkage*, *v)* una variabile locale che risieda in memoria permanente.

ESERCIZIO 8

```
INT REC_PRINT_SUM (STRUCT NODE* P) {
    IF (P -> PNEXT == NULL) {
        PRINTF ("%d\n", P->PINFO);
        PRINTF ("%d\n", 0);
        RETURN (P -> PINFO);
    }
    ELSE {
        PRINTF ("%d \n", P->PINFO);
        INT RES = REC_PRINT_SUM (P -> PNEXT);
        PRINTF ("%d\n", RES);
        RETURN (RES + P -> PINFO);
    }
}
```

ESERCIZIO 2

```
INT MAIN () {
    INT i, j, rows, COUNT = 0;
    PRINTF ("DAMMI IL NUMERO DI RIGHE\n");
    SCANF ("%d", &rows);
    FOR (i = 1; i <= 2*rows; i = i+2) {
        FOR (j = 1; j <= i; j++) {
            PRINTF ("%C", 'A'+COUNT);
            IF (j <= i/2)
                COUNT++;
            ELSE
                COUNT--;
        }
        COUNT = 0;
        PRINTF ("\n");
    }
    RETURN (0);
}
```

7. [4 punti] Scrivere cosa stampa il seguente programma, sapendo che le variabili a, b, c in *main* sono memorizzate rispettivamente agli indirizzi $0x7ffeef2c970a$, $0x7ffeef2c980a$ e $0x7ffeef2c990b$.

```
1  static int a= 4;
2  int f(int* x, int y, int* z) {
3    int res= 2;
4    if ((a++,a-=2) > 0) {
5      a ? res= (--y) + (*z+=3), (y+= *x) : 4;
6      printf("%d %d %p %p %d\n", *x, y, z, z+1, res);
7      res+= f(&y, *x, z);
8      return (res);
9    }
10   else
11     return res-1;
12 }
13
14 int main(void) {
15   int a= 4, b= 06, c= 0X3;
16   b= f(&b, a, &c);
17   printf("%d %d %d\n", a, b, c);
18 }
```

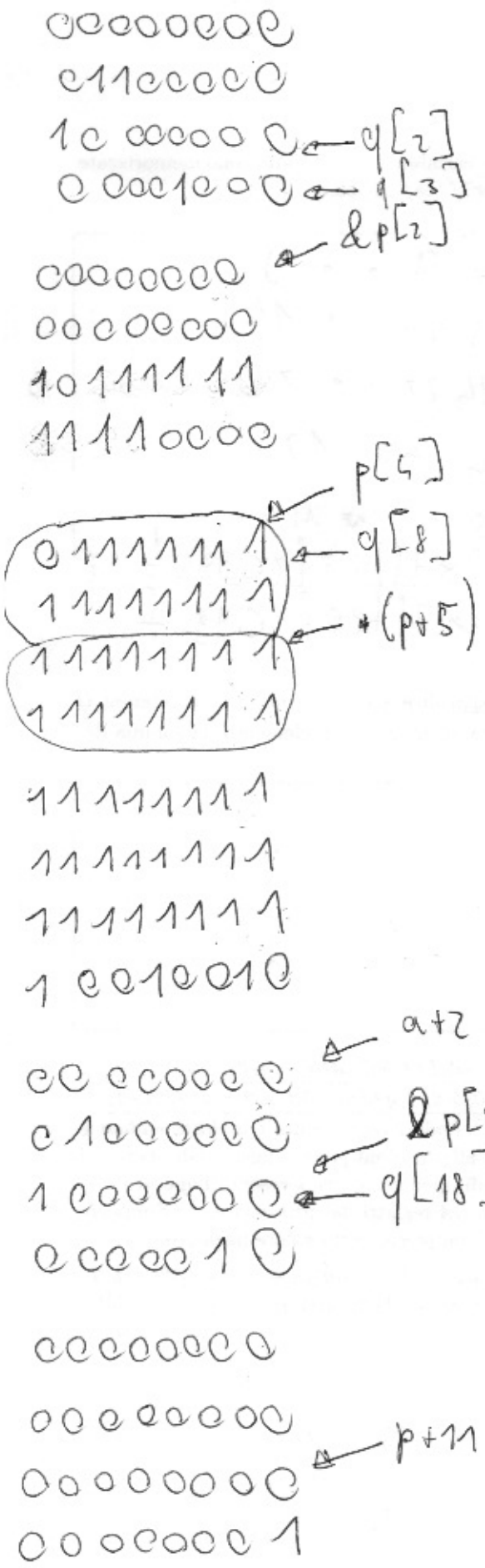
6 9 · · 9
9 14 · · 14
14 27 · · 20
4 44 12
sostituire AI · ·
0x7ffeef2c990b E
0x7ffeef2c99cf

8. [3 punti] Data la seguente *struct Node*, definire una funzione ricorsiva di nome *rec_print_sum* che stampi *i)* il campo *info* in successione dal primo all'ultimo elemento della lista, e, *ii)* per ogni elemento, la somma dei campi *info* di tutti i successivi elementi nella lista.

```
1  struct Node {
2    int info= 0;
3    struct Node* pNext= NULL;
4  }
5
```

9. [4 punti] Cerchiare le affermazioni vere dato *long long a[3]= {1536, -2, LLONG_MIN + 512};*
*short int *p = (short*) a; char *q= (char*) a; p[1]= 4097, p[3]= 4095-2, *(q+15)= 73, p[9]= 4096*4+1;*
sapendo che i tre tipi usati occupano 8, 2, e 1 byte, e $4096 = 2^{12}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Gli operatori | e & ritornano rispettivamente l'*or* e l'*and* bit-a-bit dei due operandi, mentre >> rappresenta l'operatore di *shift* di n posizioni a destra (ricordarsi che l'operazione è effettuata nei registri del processore...). Supporre che i tipi *char* siano tipi con segno: il bit più significativo uguale a 1 rappresenta un valore negativo.

A. $(q[8] \mid q[2]) + q[2]$ B. $(*(p+5) - p[4]) \ \% \ 2$ C. $(((int)(p+11) - (int)(a+2)) + q[18]) \ \% \ 7$
D. $((\&p[9] - \&p[2]) + p[8]) \ \% \ 2$ E. $(q[1] >> 2) - q[3] > 0$ F. $((q[18] << 1) \ \& \ (q[19])) == (q[20] >> 5])$



A $(q[8] \mid q[2]) + q[2] == 0$

$== \quad 1$

$11111111 == -1$

FALSA

B $*(p+5) - p[4] = 1 \;\%\; 2 == 1$

$-1 - (-2)$

VERA

C $(INT)(p+11) - (INT)(a+2) + q[18] \;\%\; 7 == 0$

$== \quad =$

$6 + 1$

FALSA

D $\&p[9] - \&p[2] + p[8] \;\%\; 2 == 1$

$==$

$7 + 512$

VERA

E q[1] NEL PROCESSORE È RAPPRESENTATO IN BIG ENDIAN

$00000110 >> 2$

$==$

$00000001 - q[3] == 15 > 0$

$== \quad ==$

$1 \quad 16$

FALSA

F q[18] E q[20] NEL PROCESSORE È RAPPRESENTATO IN BIG ENDIAN

q[18] q[20]

00000001 << 1

$==$

00000010 $\&\, q[19] \quad 00000000 >> 5$

$== \quad =$

$0 == 0$

1 VERA

q[19] portato nel processore per fare operazione &
si rappresenta (in big endian) come 01000000

ESERCIZIO 6

```
STATIC INT a = 0;
INT b = 0;
INT b;
INT c;
```

a INTERNAL LINKAGE
b EXTERNAL LINKAGE
b TENTATIVO DEFINITION CHE RIMANE DICHIARAZIONE
c EXTERNAL LINKAGE

```
VOID FUN(INT d, INT e) {
    INT f;
    STATIC INT g;
```

FUN ESTERNAL LINKAGE (FUNZIONE)
d E e NO LINKAGE
f NO LINKAGE
g NO LINKAGE E IN MEMORIA PERMANENTE

ESERCIZIO 4

SOLO LA FUNZIONE HALF_ARRAY

```
INT* HALF_ARRAY(INT* A, UNSIGNED L) {
UNSIGNED NEWL = L/2;
INT* A2 = (INT*) CALLOC (NEWL, SIZEOF(INT));
FOR(INT i=0; i<NEWL; i++)
        *(A2+i) = *(A+i) + *(A+i+NEWL);
RETURN (A2);
}
```