Prova scritta Programmazione Procedurale con Lab. - 31 Gennaio 2023 - FILA A

Nome e Cognome: _____ Matricola: _____

1. [6 punti] Su foglio protocollo, elencare le conversioni di tipo implicite, sapendo che i tipi *int* e *long int* sono rappresentati entrambi con 32 bit. Cosa stampa il programma? Quanto vale la variabile *limit* sapendo che UINT_MAX vale 4294967295?

```
1 int x = 3ULL;
2 int i = -2;
3 unsigned int limit = -100;
4 long n = 30L;
5 if ( i < limit )
6    x = limit * n;
7 printf("%d\n", x);
```

linea 1: 3ull da unsigned long long a int
linea 3: -100 da int a unsigned int
linea 4: i convertito da int a unsigned int
linea 5: i convertito da int a unsigned int
linea 6: limit da unsigned int ad unsigned long, n da long ad unsigned long
linea 6: limit * n da unsigned long a int

Il programma stampa 3 alla linea 7
limit vale -100 + (4294967295 + 1) = 4294967196

2. [6 punti] Riportare nel riquadro a fianco cosa stampa la seguente porzione di codice ricordandosi che & è l'operatore di *and* bit a bit. Evidenziare i passaggi nel foglio protocollo.

```
1 int a= 0234 - 0xa3;
2 printf("%d\n", a);
3 while ((a++ && a++) ? !!a : 0) {
4   if (++a, a++) {
5     printf("%d\n", a);
6     continue;
7   }
8   else
9   printf("%d\n", a);
10  break; }
11 !a, a= a&a;
12 printf("a: %d\n", a);
```

-7
-3
1
a: 1

3. [6 punti] Data la seguente *struct* scrivere la definizione una funzione di nome *duplica_lista* che prende come parametro una lista e ritorna un'altra lista (creata nella funzione) che contiene gli stessi valori (campo *info*) della lista passata nelle stesse posizioni.

```
1 struct Node {
2   int info;
3   struct Node* pNext;
4 }
```

4. [5 punti] Per ogni identificatore di variabile e funzione scrivere se è definito o dichiarato, ed il suo linkage.

```
1     typedef int AINT;
2     AINT a= 3;
3     int a;
4     extern int b;
5     extern int cmp(float, float);
6     static double area(double);
7
8     static int my_func(int* c) {
9       static double e= 4.5;
10      double* f= &e;
11      auto int q= 4;
12      extern int a;
13      // Altri comandi
14    }
15
```

linea 2: a definito, ext. linkage
linea 3: a tentativo definizione che rimane dichiarazione, ext. linkage
linea 4: b dichiarato, ext. linkage
linea 5: cmp dichiarato, ext. linkage
linea 6: double dichiarato, int. linkage
linea 8 : my_func definita, int. linkage
linea 8: c definito, no linkage
linea 9: e definito, no linkage
linea 10: f definito, no linkage
linea 11: q definito, no linkage
linea 12: a dichiarato, ext. linkage

5. [7 punti] Cerchiare le affermazioni vere dato *int a[5]= {1022, 131593, 131100, -1712, INT_MIN}; short int *p = (short*) a; char *q = (char*) a; q[3] += 1;* sapendo che i tre tipi usati occupano 4 , 2 byte e 1 byte, e $131072 = 2^{17}$ (valori rappresentati in *complemento a due* e *little endian*). Rappresentare la zona di memoria in cui è memorizzato l'array. A. $(p+3) > (a+1)$; B. $q[14] - q[15]$; C. $(*(q+5) \,\&\, *(q+6))\%2$; D. $((q+7) - (q+2))! = (int)(q+7) - (int)(q+2)$; E. $q[0] + *(q+10)$.

01111111	q[0]
11000000	
00000000	
10000000	
10010000	a+1
01000000	q+5
01000000	p+3, q+6
00000000	
00111000	
00000000	
01000000	q[10]
00000000	
00001010	
10011111	
11111111	q[14]
11111111	q[15]
00000000	
00000000	
00000000	
00000001	

Esercizio 3

```
struct Node* duplica_lista (struct Node* lista){
   struct Node* lista2 = NULL;
   struct Node* lista2_last = NULL;
   struct Node* pScan = lista;
   while (pScan != NULL) {
      struct Node* pNew = (struct Node*) malloc(struct Node);
      pNew -> info = pScan -> info;
      pNew -> pNext = NULL;
      if (lista2 == NULL) {
         lista2 = pNew;
         lista2_last = pNew;
       }
      else {
         lista2_last -> pNext = pNew;
         lista2_last = pNew;
      }
      pScan = pScan -> pNext;
   }
   return lista2;
}
```

A: p+3 > a+1 VERO
B: -1 - (-1) = 0 FALSO
C: l'and bit a bit ha come risultato 01000000 che vale 2 % 2 == 0 FALSO
D: la differenza tra q+7 e q+2 vale lo stesso valore (5) sia se convertita in int sia non convertita in int FALSO
E: -2 + 2 = 0 FALSO