

Prova scritta Programmazione Procedurale con Lab. - 4 Settembre 2024

Nome e Cognome: _____

Matricola: _____

1. 5 punti Riscrivere il seguente programma aggiungendo opportune operazioni di conversione esplicite (*cast*) ovunque sia presente una conversione implicita, in modo da non generare possibili *warning* di conversione. Inoltre, dati $UINT_MAX = 4294967295$ e $USHRT_MAX = 65535$, scrivere i valori di a e d .

<pre> 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 }</pre>	<pre> unsigned a = (unsigned int) -1; float b = (float) pow2((float) a); double *p = (double*) malloc(sizeof(double)); unsigned short d = (unsigned short) -4; a vale 4294967295 d vale 65532</pre>
--	--

2. 7 punti Riportare cosa stampa il seguente programma, sapendo che b si trova all'indirizzo $0x1047e601d$.

```

1 int a = 0x19, b= 017;
2
3 void f(void) {
4     while (b > 7? b-=2 : b) {b--; ++a;}
5 }
6
7 int main() {
8     printf("%p\n", &b);
9     for (int i= 1 ; f(), b= b+i; i++, printf("%d
10         %d\n", a, b))
11         if (i > 4)
12             break;
13     else
14         do {(&a)++;} while (!a && a++);
15     printf("%p", ((long long*) &b) +1);
16 }
```

```

0x1047e601d
35 1
37 2
40 3
44 4
0x1047e6025
```

3. 6 punti Data la seguente struttura che rappresenta ogni nodo di una lista, definire una funzione di nome *duplicate* che prende come parametro *int infoToDuplicate*. La funzione aggiunge un nuovo elemento con valore *infoToDuplicate* in testa alla lista solo se tale lista contiene già un elemento con valore identico a *infoToDuplicate*. Considerare un puntatore con visibilità globale alla lista di nome *pFirst*.

```

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

4. 5 punti Per ogni identificatore di variabile e funzione riportare se definizione o solo dichiarazione, ed il linkage.

```

1 #define PI 3.14159
2 static double a= 4.2;
3 extern float area(double b);
4 typedef int c;
5 c d= 1;
6
7 extern int myFunc(float e, int* f) {
8     static float g= 4.1;
9     int h=5, i[*f];
10    extern c d;
11    // Comandi...
12 }

```

```

linea 2: a definito int link
linea 3: area dichiarato ext link
linea 3: b dichiarato no link
linea 5: d definito ext link
linea 7: myFunc definito ext link
linea 7: e e f definite no link
linea 8: g definito no link
linea 9: h e i definite no link
linea 10: d dichiarato ext link

```

5. 7 punti Rappresentare la mappa di memoria relativa all'array $int\ a[5] = \{25, [2] = INT_MAX, 131046, 131328\}$; $short\ int\ *p = (short*)\ a$; $char\ *q = (char*)\ a$; e stabilire il valore di verità dei seguenti quesiti sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e $131072 = 2^{17}$ (valori rappresentati in *little endian* e complemento a due). A. $((\&a[4] - a)\%2) == 0$ B. $(int)(a + 5) - (int)(p + 6) + q[5] == 2$ C. $p[2] || *(q + 10)$

Esercizio 3

```

void duplicate(int infoToDuplicate) {
    if (pFirst == NULL) {
        printf("Empty list\n");
        return;
    }

    // Controllo se infoToDuplicate è dentro la lista isIn = 0;
    Node* pScan = pFirst;
    do {
        if (pScan -> info == infoToDuplicate) {
            isIn = 1;
            break;
        }
        pScan = pScan->pNext;
    } while(pScan!= NULL);

    if (isIn) {
        Node *pNew = (Node*) malloc(sizeof(Node));
        pNew->info = infoToDucuplicate;
        pNew->pNext= NULL;
        if(pFirst == NULL)
            pFirst = pNew;
        else {
            pNew-> pNext= pFirst;
            pFirst= pNew; }
    }
}

```

ES 9

$x == p == q$

\swarrow \nwarrow $\neq p$
 10011000
 00000000
 00000000
 00000000

\swarrow $p[2]$
 00000000
 00000000 $q[5]$
 00000000
 00000000

$\# (p+4)$
 11111111 $q[8]$
 11111111 $q[9]$
 11111111 $q[10] \neq (q+10)$
 11111111 $q[11]$

\swarrow $p+6$
 01100111
 11111111
 10000000 $\# (q+14)$
 00000000

\swarrow $a[4]$
 00000000
 10000000 $q[17]$
 01000000 $\# (q+18)$
 00000000

\swarrow $a+5$

A: $4 \% 2 == 0$ VERO

B: $20 - 12 + 0 == 0$ FALSO

C: $0 \parallel -1$