Prova scritta Programmazione Procedurale con Lab. - 27 Giugno 2023

Nome e Cognome: _____ Matricola: _____

1. [5 punti] Elencare le conversioni di tipo (... *da* ... a). Dato *USHRT_MAX = 65535*, scrivere il valore finale della variabile *b* sapendo che il carattere *a* ha valore 97 e le altre lettere seguono in ordine alfabetico.

```
1   long int fun2(long int p) {
2     return (p + 'c'); }
3
4   int fun1(int p) {
5     char a= 'a';
6     return fun2(p - a); }
7
8   int main(void) {
9     unsigned short a =  -3;
10    float b= fun1(a);
11  }
```

per la regola (slide 18 conversioni) n + (USHRT_MAX + 1) ad a viene assegnato
-3 + (65535 + 1)= 65533
b= 65533 - 97 + 99= 65535.0

linea 9:- 3 da int a unsigned short
linea 10: a da unsigned short a int
linea 5: 'a' da int a char
linea 6: a da char a int in p-a
linea 6: p-a da int a long int in chiamata fun2
linea 2: 'c' da int a long int in p+c
linea 6: risultato fun2 da long int a int (ritorno fun1)
linea 10: risultato fun1 da int a float

2. [6 punti] Scrivere cosa stampa la seguente porzione di codice.

```
1   int a= 0xfb;
2   while(a > 021 ? !!1: !!(a--, --a)
    ) {
3     printf("%d \n", a);
4     if (a + 2 >= 0x10) {
5       a= -0x2;
6       continue; }
7     a+= 3; }
8   !(a+1) && a++;
9   printf("a: %d\n", a);
10
```

251
-4
-3
-2
-1
a: 0

3. [6 punti] Dire quali compilazioni provocano errore a causa del linker (e perché): 1) *gcc –o write write.c*, 2) *gcc –c main.c*, 3) *gcc –o main main.c*, 4) *gcc write.c main.c -o main*. In caso il punto 4) ritorni un errore, descrivere come può essere corretto. Infine, che tipo di *linkage* hanno *count*, *i*, e *write*? Cosa stampa il programma *output*? Elencare tutte le definizioni e dichiarazioni in ogni file.

main.c

```
void mywrite(int *count);
extern int count;

int main(void) {
  do {
    mywrite(&count);
  }while(count >=0);
}
```

write.c

```
#include <stdio.h>

extern int i= 1;
int count= 4;
void mywrite(int *a) {
  static int count= -5;
  (*a)--;
  printf("%d\n", count= count - i);
}
```

1) errore: manca la definizione di main
3) errore: manca la definizione di mywrite e count
count (globale), i, mywrite hanno linkage esterno in tutti e due i file
count (locale a mywrite) ha no linkage
in main.c definiti: main
in main.c dichiarati: mywrite, count
in write.c definiti: i, count (globale), mywrite, count (locale in mywrite), a
in write.c dichiarati: mywrite, count

Stampa
-6
-7
-8
-9
-10

4. [6 punti] Data la seguente struttura, definire una funzione di nome *ins_head* che prende come parametri un puntatore a *struct Node* (il puntatore al primo elemento della lista) e un valore *int*, e inserisce un nuovo elemento in testa alla lista (cioè come primo elemento), con il campo *info* assegnato al valore passato. La funzione non crea il nuovo elemento e stampa un errore, se il campo *info* del nuovo elemento è minore del campo *info* dell'elemento successivo in lista.

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

5. [7 punti] Cerchiare le affermazioni vere dato *int a[5]= {INT_MAX - 7, 1287, INT_MIN + 528, -10, 312}; short int *p = (short int*) a; char *q= (char*) a; p[3]= SHRT_MAX, p[5]+= 2048, q[18]= ∼ q[19];* sapendo che i tre tipi usati occupano 4, 2, e 1 byte, con valori rappresentati in *little endian* e complemento a due. Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). Cerchiare le affermazioni vere qui sotto.

(A.) $(((int)(q+13) - (int)(a+1)) - q[1]) \,\%\, 3$ B. $(*(p+7) \,\&\, p[9]) \,\%\, 5$ (C.) $\sim (\&p[5] - \&p[1] - q[9] * 2)$

Esercizio 5

00011111
11111111
11111111
11111110

11100000
10100000
11111111
11111110

A: (int)(q+13) - (int) (a+1) == 9 q[1] == -1
9 - (-1) % 3 == 1 VERA

B: 11111111 11111111 == *(p + 7)
11111111 00000000 == p[9]
in and bit a bit il risultato è p[9] che
equivale a 255%5 == 0 FALSA

00001000
01000000
00000000
00010001

01101111
11111111
11111111
11111111

00011100
10000000
11111111
00000000

C: &p[5] - &p[1] == 4 (restituisce il numero di short int tra i due puntatori) q[9] * 2 == 4
&p[5] - &p[1] - (q[9] * 2) == 0, che negato bit a bit porta a VERA

Esercizio 4

```
void ins_head(struct Node** pFirst, int infoField) {
 if (*pFirst != NULL)
  if (((*pFirst) -> pInfo) < infoField) {
   printf("Errore parametro infoField minore del successivo elemento")
   return;
  }
  else {
   struct Node *pNew = (struct Node*) malloc(sizeof(struct Node));
   pNew-> pNext= NULL;
   pNew-> pInfo= infoField;
   if(*pFirst == NULL)
     *pFirst = pNew;
   else {
    pNew-> pNext= *pFirst;
    *pFirst= pNew;
   }
  return;
 }
}
```