

Prova scritta Programmazione Procedurale con Lab. - 15 Gennaio 2024

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 (in ASCII) e le altre lettere seguono in ordine alfabetico.

```

1 long long int fun2(int p) {
2     return (p + 'g' - 'a'); }
3
4 int fun1(long p) {
5     char a= 'i';
6     return fun2(p - a + 'b'); }
7
8 int main(void) {
9     unsigned short a = -2L;
10    float b= fun1(a);
11    printf("%f\n", b);
12 }
```

linea 9: -2L da long a unsigned short
 linea 10: a da unsigned short a long
 linea 5: 'i' da int a char
 linea 6: a da char a long, 'b' da int a long
 linea 6: p-a-'b' da long a int
 linea 2: p + 'g' - 'a' da int a long long (ritorno funzione fun2)
 linea 6: risultato fun2 da long long a int (ritorno funzione fun1)
 linea 10: risultato fun1 da int a float

b vale alla fine 65533.0 (printf)

linea 9: a è inizializzato a 65534 , -2 + (USHRT_MAX + 1)

2. 5 punti Scrivere cosa stampa il seguente programma e motivare su foglio protocollo.

```

1 int a= 0xfb5 - 07674;
2 printf("%d\n", a);
3 while(-a % 2 ? (a++, a + 1): !!1) {
4     printf("%d \n", a);
5     if (!a) {
6         a= 0xf;
7         break; }
8     a+=1; }
9 !(a+1) && a++;
10 printf("a: %d\n", a);
```

-7
 -6
 -4
 -2
 0
 a: 15

3. 6 punti Data la seguente *struct Node* definire su foglio protocollo una funzione di nome *inserisci_3_posizione()* che inserisce un nuovo valore intero passato come parametro sempre in terza posizione, se la lista ha già almeno due elementi (gestire se la lista ha meno elementi): se la lista originale è 7-4-11-18 e il valore passato è 5, la nuova lista sarà 7-4-5-11-18. Supporre un puntatore ad inizio lista globale di nome *pFirst*.

```

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

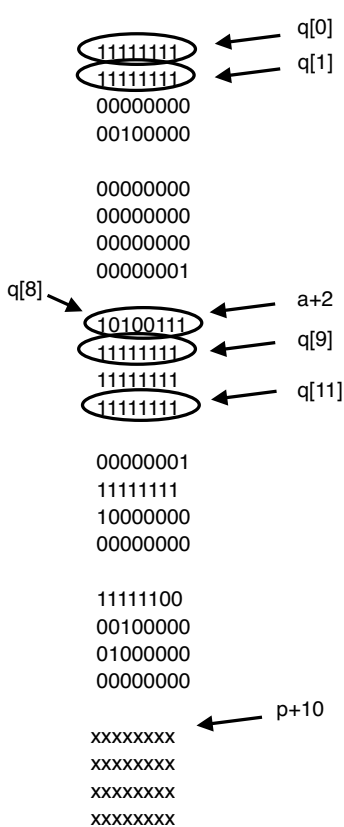
4. 5 punti Per ogni identificatore di variabile e funzione scrivere se è definito o dichiarato, ed il suo linkage.

```

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

linea 2: a definita, external linkage
 linea 3: a tentativo di definizione che rimane dichiarazione, external linkage
 linea 4: b dichiarata, external linkage
 linea 5: c definita, external linkage
 linea 6: cmp dichiarata, external linkage
 linea 8: my_func definita, external linkage
 linea 8: d definita, no linkage
 linea 9: e definita, no linkage
 linea 10: f definita, no linkage
 linea 11: q definita, no linkage
 linea 12: b dichiarata, linkage esterno

5. 7 punti Cerchiare sotto le affermazioni vere dato $\text{int } a[5] = \{INT_MAX, INT_MIN, -27, 130944, 132159\}$; $\text{short int } *p = (\text{short}*) a$; $\text{char } *q = (\text{char}*) a$; $*(p+1) = 1024$; 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). Rappresentare la mappa di memoria e giustificare perché le affermazioni sono vere o false. A. $((\text{int})(p+10) - (\text{int})(a+2) + q[1]) \% 2$
 B. $q[0]! = q[9]$ C. $!((a+2) - (a+3))$ D. $*(q+8) \wedge *(q+11) > 25$
6. 2 punti Quali tra queste keyword permettono di dichiarare una variabile (senza definirla definirla).
 auto; register; main; static; extern; declare.



Esercizio 6

- A: $(12 + (-1)) \% 2$ VERO
 B: $-1 != -1$ FALSO
 C: $!(-1)$ FALSO
 D: $10100111 \wedge 11111111 = 01011000$ che vale 26
 $26 > 25$ VERO

Esercizio 3

```
void inserisci_3_posizione (int nuova_info) {
    if (!(pFirst != NULL && pFirst -> pNext != NULL)) {
        printf("La lista ha meno di 2 elementi, non inserisco %d", nuova_info);
        return;
    }

    struct Node* nuovo_elem = (struct Node*) malloc(sizeof(struct Node));
    nuovo_elem -> info = nuova_info;
    nuovo_elem -> pNext = NULL;
    struct Node* pSecond = pFirst -> pNext;
    struct Node* pThird = pSecond -> pNext;
    pSecond -> pNext = nuovo_elem;

    if (pThird != NULL)
        nuovo_elem -> pNext = pThird;
}
```