

Prova scritta Programmazione Procedurale con Lab. - 30 Gennaio 2024

Nome e Cognome: _____

Matricola: _____

1. 6 punti Riportare le conversioni di tipo -implicite- (*valore da tipo a tipo*) e scrivere cosa stampa il programma, sapendo che *UINT_MAX* vale 4294967295 e *CHAR_MAX* vale 127.

```

1 int x = 0, i = -2;
2 char a= (char) 80, b= (char) 80, c= (
  char) 100;
3 a= (a*b) / c;
4 unsigned int limit = 10U;
5 long n = 20L;
6 if ( i < limit )
7   x = limit * n;
8 printf("%d %d\n", a, i);

```

linea 3: a, b, c convertiti da char a int (integer promotion)
 linea 3: risultato di (a*b) / c convertito da int a char
 linea 6: i convertito da int ad unsigned int (regola 1 slide)
 linea 7: n e limit convertiti entrambi a unsigned long (se long e int entrambi 32 bit, regola 2 slide)
 linea 7: il risultato limit * n convertito da unsigned long a int

stampa 64 -2 (%d stampa il valore di i come intero)

2. 5 punti Su foglio protocollo, scrivere la definizione di una funzione *crea_array()* che ha un parametro di nome *lunghezza* (di tipo *unsigned int*). Questa funzione crea un array con quella lunghezza, e riempie tutti i valori di tale array leggendoli da tastiera, utilizzando l'aritmetica dei puntatori, infine, restituisce l'array come valore di ritorno.

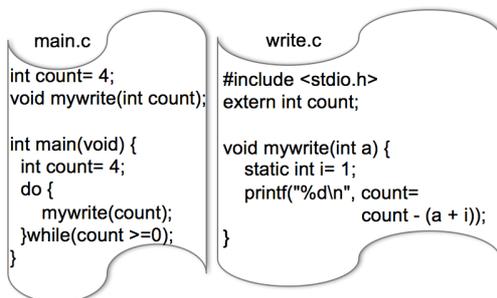
3. 6 punti Data la seguente *struct Node*, definire su foglio protocollo una funzione di nome *inserisci_posizione()* che prende come parametri un valore intero (*int*) e un valore senza segno (*unsigned int*), che rappresentano rispettivamente il valore da inserire nella lista e la posizione dove inserirlo. Per esempio, se la lista originale è 7-4-11 e i valori passati sono -2 e 2, la nuova lista sarà 7--2-4-5-11. Supporre un puntatore ad inizio lista globale di nome *pFirst*.

```

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

```

4. 6 punti Dire quali comandi di compilazione provocano errore a causa del linker (e perché): 1) *gcc -c write.c*, 2) *gcc -o main main.c*, 3) *gcc -o write write.c*, 4) *gcc -c main.c*, 5) *gcc write.c main.c -o main*. In caso il punto 5) ritorni un errore, descrivere come possa essere corretto. Dopo tutti questi passi, che tipo di *linkage* hanno *count*, *i*, e *mywrite*, ed in quale file sono definite? Cosa stampa il programma?



2) *mywrite* non definita, 3) funzione *main* e *count* non definiti
count ext in entrambi i file
mywrite ext in entrambi i file
i ha no linkage

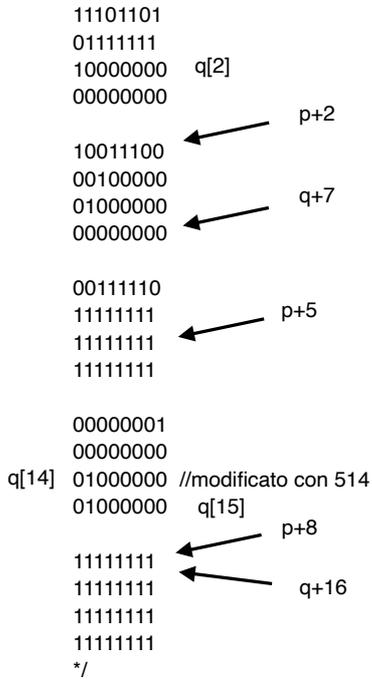
count definito in *main.c*
mywrite definito in *write.c*
i definito in *write.c*

Stampa
 -1
 -6
 -11
 -16 continua

Nessun errore al punto 5) da correggere

5. 7 punti Cerchiare sotto le affermazioni vere dato $int\ a[5]= \{130743, 132153, -132, CHAR_MIN, UINT_MAX\};$ $short\ int\ *p = (short*)\ a;$ $char\ *q= (char*)\ a;$ $*(p+7) = 514;$ 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. $((int)(q+16) - (int)(p+2) + q[2]) / 2 == 7.5$
 B. $q[14] \& q[15] != 1$ C. $!((a + 2) - (a + 3))$ D. $*(p + 5) - *(p + 8)$

Esercizio 5
 mappa di memoria



Esercizio 2
 Se scegliamo un array di interi:

```

int* crea_array(unsigned int lunghezza) {
    int* my_array= (int*) calloc(lunghezza, sizeof(int));
    for (int i=0; i < lunghezza; i++)
        scanf("%d", (my_array + i));
    return my_array;
}
  
```

Esercizio 3

```

void inserisci_posizione (int nuova_info, unsigned int posizione) {
    unsigned int counter = 0;
    struct Node* pScan= pFirst;
    if (pScan != NULL)
        while (pScan -> pNext != NULL) {
            pScan = pScan -> pNext;
            counter++;
        }
    if (posizione > counter + 1) {
        printf("Lista con %u elementi, non posso inserire\n", counter);
        return;
    }
    else {
        struct Node* pNew= (struct Node*) malloc(sizeof(struct Node));
        pNew -> info= nuova_info;
        pNew -> pNext= pNew;
        pScan= pFirst;
        if (posizione == 1) {
            pFirst = pNew;
            pFirst -> pNext = pScan;
        }
        else {
            for (int i=1; i < posizione - 1; i++)
                pScan = pScan -> pNext;
            struct Node* pTemp = pScan -> pNext;
            pScan -> pNext = pNew;
            pNew -> pNext = pTemp;
        }
    }
}
  
```

tutte false

A: $12 + q[2] == 13$ che diviso due fa 6, quindi diverso da 7.5 FALSO

B: si esegue con la seguente precedenza $q[14] \& (q[15] != 1)$, dato che gli operatori di di relazione hanno precedenza maggiore a quelli logici quindi $01000000 \& 10000000 == 0$ FALSO

C: $5 : *(q+7) : *(q+2)$, dato che 5 e' vero, $*(q+7) == 0$ quindi FALSO

D: $-1 - (-1) == 0$ FALSO