

Prova scritta Programmazione Procedurale - 1 Febbraio 2022 - FILA A

Nome e Cognome: _____

Matricola: _____

1. **6 punti** Elencare le conversioni di tipo implicite (... da ... a). Supporre che *int* e *long* siano entrambi rappresentati su 32 bit, mentre *long long* sia 64, e che `LLONG_MAX` abbia tipo *long long* e `UINT_MAX` abbia tipo *unsigned int*. Scrivere il valore finale della variabile *x*.

```
1 int x = 0;
2 int i = -1L;
3 unsigned int limit = 200U;
4 long n = 5UL;
5 if ( i < limit )
6     x = limit * n;
7 else
8     x = (LLONG_MAX-UINT_MAX)+1;
```

linea 2: -1L da long int a int
 linea 4: 5UL da unsigned long a long
 linea 5: i convertito da int ad unsigned int (regola 1 slide conversioni)
 linea 6: sia n che limit vengono convertiti a unsigned long (regola 2 slide conv.)
 linea 6: risultato espressione convertito da unsigned long a int
 linea 8: `UINT_MAX` convertito da unsigned long a long long (regola 2 slide conv.)
 linea 8: 1 convertito da int a long long
 linea 8: risultato espressione convertito da long long a int

Il valore finale della variabile x è 1 (stesso esempio in slide conversioni)

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

```
1 int a= 0xaa - 0161;
2 printf("%d\n", a);
3 for (; a--, a--? a-1 : a-2; a + 1) {
4     printf("%d \n", a);
5     if (a > 0x32) {
6         a--;
7         continue;
8         printf("%d\n", a);
9     }
10 else
11     break; }
12 !(a) || a++;
13 printf("a: %d\n", a);
```

57
 55
 52
 49
 a: 50

3. **6 punti** Data la seguente *struct* definire una funzione di nome *cancella_secondo* che cancella l'elemento della lista in seconda posizione (sempre se esiste); supporre che il puntatore ad inizio lista *pfirst* sia globale (non passarlo come parametro).

```
1 struct Node {
2     int info= 0;
3     struct Node* pNext= NULL;
4 }
5
```

4. **5 punti** Su foglio protocollo, scrivere un solo file con un nome a piacere (che non deve contenere la funzione *main*) in cui siano presenti *i*) due tentativi di definizione di cui uno rimane dichiarazione e l'altro diventa definizione, *ii*) un identificatore di variabile globale con *internal linkage* (definita), *iii*) due identificatori con *external linkage* (di cui un identificatore di funzione e uno di variabile, entrambi definiti), *iv*) una variabile *no linkage* e definita, che risieda in memoria permanente, *v*) una variabile *no linkage* solamente dichiarata (e non definita). Evidenziare chiaramente questi cinque punti. Scrivere il comando *gcc* per compilare il file senza nessun tipo di errore.

5. **7 punti** Cerchiare le affermazioni vere dato `int a[5]= {73,-10,65523,INT_MAX,INT_MIN};` `short int *p = (short*) a;` `char *q = (char*) a;` sapendo che i tre tipi usati occupano **4, 2 e 1** byte, e `65536 = 216` (valori rappresentati in *complemento a due e little endian*). Rappresentare la zona di memoria in cui è memorizzato l'array. A. `q+3 > &a[1];` B. `*(q+15)-q[0] < 55;` C. `!*(q+8);` D. `(&p[3]-&p[1])%2 == 0`

Esercizio 3

```
void cancella_secondo (void) {
    if (pfirst == NULL) {
        printf("Lista vuota\n");
        return;
    }
    if (pfirst -> pNext == NULL) {
        printf("La lista ha un solo elemento\n");
        return;
    }
    struct Node* terzo = pfirst-> pNext-> pNext;
    free(pfirst -> pNext);
    pfirst -> pNext = terzo;
}
```

10010010	q[0]
00000000	
00000000	&p[1]
00000000	q + 3
01101111	&a[1]
11111111	
11111111	&p[3]
11111111	
11001111	*(q + 8)
11111111	
00000000	
00000000	
11111111	
11111111	
11111111	
11111110	*(q + 15)
00000000	
00000000	
00000000	
00000001	

Esercizio 4

```
int a=5;
int a; //tentativo definizione, rimane dichiarazione

int b; //tentativo definizione, diventa definizione, ha external linkage

static int c=0; //internal linkage

void pippo(void) { //pippo external linkage e definita
    static int d=0; //no linkage, definita e memoria permanente
    extern int a; //locale solamente dichiarata
}
// gcc -c file.c
```

Esercizio 5

- A Falso: $q + 3$ è prima di $\&a[1]$ come indirizzo in memoria
 B Vero: $127 - 73 = 54 < 55$
 C Falso: $*(q+8)$ è un valore negativo, negato vale 0 quindi falso
 D Vero: tra i due indirizzi ci sono 2 short (4 byte)