

Università Ca' Foscari
Corso di Laurea in Informatica

Programmazione part-time

Esame

Nome: _____

Matricola: _____

Andrea Marin , a.a. 2010/2011

Teoria. (10 punti) Rispondere ai seguenti quesiti utilizzando eventualmente gli appositi spazi bianchi.
Tempo totale a disposizione: 20 min.

(2^{pts}) **1.** Dato il seguente codice:

```
int int* b;  
int c[]={2,4,6};  
b = c+1;  
b[0] = c[0];  
b[1] = b - c;
```

qual è il contenuto del vettore **c** al termine dell'esecuzione del codice (scrivere *codice errato* se si ritiene il codice scorretto)?

2 pts

(3^{pts}) **2.** Dato il seguente programma:

```
char* crea_stringa() {  
    char saluto[] = "Hello world!";  
    return saluto;  
}  
  
int main(){  
    char* s = crea_stringa();  
    s[0]='h';  
    printf("%s", s);  
}
```

Si spieghi l'errore e si riscriva il codice correggendolo.

3 pts

(3^{pts}) **3.** Date le seguenti dichiarazioni:

```
int a,b;  
int *c;;  
int **d,**e;
```

si stabilisca per le seguenti scritture:

3 pts

- se identificano una variabile (possono comparire a sinistra di un assegnamento) (*in caso affermativo barrare la casella della colonna A*)
- il tipo (*specificarlo nella colonna B*)
- se é presente un errore che non consentirebbe la compilazione (*in tal caso barrare la colonna C e lasciare in bianco le colonne A e B*)

	A	B	C
<code>*d-*e</code>			
<code>d - e</code>			
<code>*d + *e</code>			
<code>**d + **e</code>			
<code>*(c+5)</code>			
<code>** (e+3)</code>			
<code>c-5</code>			

(2^{pts}) 4. Data la seguente funzione:

```

void copia_stringa(char* s1, char* s2) {
    s1 = (char*) malloc (strlen(s2)+1);
    strcpy(s1,s2);
}

int main(){
    char[] saluto="Hello world!";
    char* copiasaluto;
    copia_stringa(copiasaluto, saluto);
    printf("%s" copiasaluto);
    free(copiasaluto);
}
    
```

2 pts

Si spieghi l'errore e si riscriva il codice mantenendo la copia della stringa allocata nell'heap.

Pratica. (23 punti) Nello svolgimento del seguente esame, il candidato crei una cartella con il proprio cognome e numero di matricola e la lettera c (e.g. Rossi887766c) inserendo all'interno i file corrispondenti agli esercizi che si intendono consegnare (e.g. Esercizio1.c). Le prime righe del file devono essere dei commenti che specifichino il vostro nome e cognome e l'esercizio a cui si riferiscono.

Tempo a disposizione: 1h e 45 min.

- (6^{pts}) **1.** Scrivere una funzione C che crei una lista di $n \leq 10$ interi casuali tra 1 e 10 (n sia un parametro) tutti fra loro diversi. Una cella deve contenere un numero diverso da tutti gli altri e tutti i numeri leciti devono avere uguale probabilità di essere scelti. Per estrarre un numero casuale da 1 a 10 `rand()%10 + 1`.

6 pts
- (17^{pts}) **2.** Si vuole scrivere un programma C per al fine di facilitare la partecipazione al gioco dello *Scarabeo* in versione semplificata. Il programma manterrà in una lista di stringhe un elenco di parole che chiameremo **dizionario**. Per ogni parola si può calcolare un punteggio come segue:
- Le vocali *a, e, i* valgono un punto
 - Le vocali *o, u* valgono due punti
 - Le consonanti *z, h* valgono otto punti
 - Tutte le altre lettere valgono tre punti
- (a) (2 pts) Definire il tipo dati `t_dizionario` per la memorizzazione del dizionari di parole.
- (b) (3 pts) Scrivere la funzione `int calcola_punti(char* str)` che calcoli il punteggio associato ad una stringa.
- (c) (3 pts) Scrivere la funzione `int match(char* str, char* pattern)` che restituisca `1` se la stringa `str` soddisfa le specifiche della stringa `pattern`, `0` altrimenti. La stringa `pattern` contiene delle lettere e il carattere speciale `?`. La stringa `str` soddisfa il `pattern` se le lettere nella stessa posizione sono uguali oppure `pattern` in quella posizione ha un `?`. Per esempio *cicala* soddisfa il pattern *c??al?* ma non il pattern *c?al?*.
- (d) (4 pts) Scrivere la funzione `char* seleziona(t_dizionario dizionario, char* pattern)` che restituisca la stringa con massimo punteggio che soddisfa il `pattern`.
- (e) (5 pts) Scrivere la funzione `char* seleziona(t_dizionario dizionario, char* pattern)` che restituisca la stringa con massimo punteggio che soddisfa il `pattern` anche eccedendo la lunghezza. Ad esempio il pattern *c?cal?* è soddisfatto da *cicala* ma non da *cicalino* secondo le definizioni precedenti. In questo esercizio, invece, va restiuito *cicalino* perchè di maggiore punteggio.
- | |
|--------|
| |
| 17 pts |