Scuola Superiore Sant'Anna
di Studi Universitari e di Perfezionamento

Esame di Fondamenti di Informatica
Corso interno

27/05/2008

Cognome: ______________________________

Nome: ______________________________

Regole:

Per le prime 11 domande,

- una risposta esatta fa ottenere 4 punti,
- una risposta sbagliata vale -1 punto,
- nessuna risposta, vale 0 punti.

L'ultima domanda richiede la scrittura di una semplice funzione C. Vale fino ad un massimo di 6 punti, a seconda della correttezza, non sono previsti punti negativi in questo caso.

Il test viene considerato superato se si superano i 30 punti complessivi.

1. Consider the following code: which one of the sentences is true?

```
int a; char y;
...
y = (char) a;
```

(a) y is transformed into an integer (b) The code produces a run-time error
(c) The code produces a syntax error (d) y is assigned the truncated value of variable a

2. Consider the following code: which one of the sentences is surely wrong? (i.e. it is wrong in all possible cases)

```
int a; char y;
...
a = *((int *)&y);
```

(a) y is assigned a random value
(b) The code produces a syntax error
(c) the code produces a run-time error
(d) a is assigned the value of y interpreted as an integer

3. What is the value of variable p after the following instructions?

```
int a[10] = {1, 2, 3, 4, 5, 8, 9, 10, 12, 15};
int insert(int x, int a[]) {
  int i;
  for (i=0; i < 10; i++) {
    if (x > a[i]) continue;
    else break;
  }
  if (i<10) a[i] = x;
  return i;
}
int p; int x = 11;
p = insert(x, a);
```

(a) p = 10 (b) p = 8 (c) p = 11 (d) none of the previous

4. What is the value of variable x after the following instructions?

```
int check(char *s) {
  int i = 0;
  while (*s != 0) {
    if (*s == 'r') return i;
    ++i; ++s;
  }
  return -1;
}
char mys[] = “abracadabra”;
int x = check(mys + 3);
```

(a) x = 0 (b) x = 6 (c) x = -1 (d) x = 2

5. What is the output printed by the following code segment on screen?

```
int spec(char *s) {
    int n = strlen(s);
    int i = 0;
    while(i < n/2 && s[i]==s[n-i-1]) ++i;
    if (i == n/2) return 1;
    else return 0;
}
printf(“%d %d %d\n”, spec(“abcdabcd”), spec(“abccba”), spec(“abcdcba”));
```

(a) 1 0 1 (b) 0 1 1 (c) 0 1 0 (d) 1 1 1

6. For which values of b variable k goes into overflow?

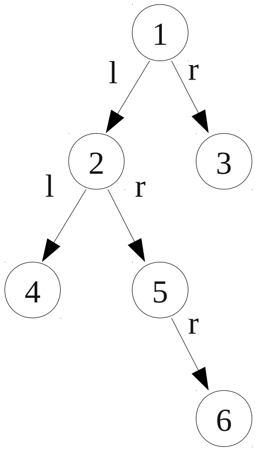
```
int k = b;
while (k<10) k = 3*k-2;
```

(a) b > 3 (b) b > 1 (c) b > 2 (d) b <= 1

7. Consider the binary tree shown on the right. What is the output of function print of such a tree?

```
struct node {
 int data;
 struct node *r, *l;
} root;

void print(struct node *n) {
 if (n==0) return;
 print(n->l);
 printf("%d ", n->data);
 print(n->r);
}
```



(a) 4 2 5 6 1 3 (b) 6 4 5 2 3 1
(c) 1 2 3 4 5 6 (d) 6 5 4 3 2 1

8. Consider the implementation of an ordered set (e.g. a phone book) by a balanced binary tree and by an ordered linked list. Which of the following sentences is surely wrong?

(a) The balanced binary tree allows a fast search of elements
(b) Insertion and deletion of elements is more efficient with the double linked list
(c) Listing all elements has the same complexity in the double linked list and in the binary tree
(d) The double linked list is more efficient from a memory consumption point of view

9. What is the last number printed by the following code segment?

```
s = 0; i = 0;
while (i < 100) {
  printf("%d\n", s);
  s = s + i * 2;
  ++i;
}
```

(a) 9000 (b) does not finish (c) 9900 (d) none of the previous

10.Assuming that x >= 0, which of the statements below is **false** after the code is executed?

```
y=1;
while (y<=x) y = y * 2;
```

(a) y may be equal to x+1 (b) y must be a power of 2

(c) y may be odd (d) y may be equal to x

11.Consider the following code. Which one of the following sentences is true?

```
int find(int d) {
 if (d%3 == 0) return d;
 else if (d%3 == 1) return find(d+4);
 else if (d%3 == 2) return find(d-4);
}
find(10);
```

(a) the function call returns 10 (b) the code produces “out of memory”

(c) the function call returns 6 (d) the code does not compile

12.Given the following data structure (array of strings):

```
struct phone_book {
 char *book[1000];
 int nelem;
} my_phone_book;
```

where book is an array of strings (pointer to char) and nelem is a variable that store how many elements are in the array.
Write a function find() with the following prototype:

```
 int find(char *str, struct phone_book *b);
```

that returns -1 if the string str is not present in the array; or the index of the array where str is stored.
To compare two strings, you can use the library function:

```
   int strcmp(const char *s1, const char *s2);
```

that compares s1 and s2 according to the alphabetic order, and returns **-1** if s1 comes before s2, **0** if s1 is the same as s2, **1** is s1 comes after s2.