

Fall term 2010  
KAIST EE209 Programming Structures for EE

## Mid-term exam

Thursday Oct 21, 2010

Student's name: \_\_\_\_\_

Student ID: \_\_\_\_\_

The exam is closed book and notes. Read the questions carefully and focus your answers on what has been asked. You are allowed to ask the instructor/TAs for help only in understanding the questions, in case you find them not completely clear. Be concise and precise in your answers and state clearly any assumption you may have made. All your answers must be included in the attached sheets. You have 70 minutes to complete your exam. Be wise in managing your time.

Please do not fill in the "Score" fields below. Self-grading is not allowed. Good luck!

### Scores

Question 1	_____ /10
Question 2	_____ /10
Question 3	_____ /15
Question 4	_____ /20
Question 5	_____ /15
Total	_____ /70

1. Number Systems (10 pts)

a) `5 < 10 < 20;`

Is this C expression true, false or undefined? (2pts)

b) `1 && 2 && 3 || 0;`

Is this C expression true, false or undefined? (2pts)

c) `i = j = 2;`

Is this C expression true, false, or undefined? (2pts)

d) `char ch = -5;`

Write ch as an 8-bit binary number. (4 pts)

2. (String functions) Please fill out the function bodies below. (10pts)

`int mystrlen(const char *s)` returns the length of the string `s`, not including the terminating `'\0'` character. That is, the behavior is exactly the same as `strlen(const char* s)` in C run-time library (except the return value: `strlen()` returns `size_t` but we use `int`) (5pts)

```
int mystrlen(const char *s)
{
```

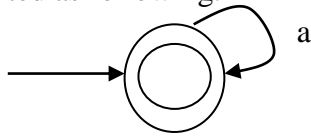
```
}
```

`char *mystrdup(const char *s)` returns a pointer to a new string which is a duplicate of the string `s`. Please allocate the memory for the new string with `malloc()`. The behavior is the same as `strdup(const char *s)` in C run-time library. You can **use** `mystrlen()` but **cannot use** `strcpy()` in the body of `mystrdup()`. (5pts)

```
char *mystrdup(const char *s)
{
```

```
}
```

3. (DFA) It is known that every regular expression can be described by a deterministic finite automata (DFA). Assuming we have characters 'a', and 'b' for input,  $^a^*$  can be represented as following.



Please draw your DFA diagrams with a start state (marked with an incoming arrow) and success states(circled twice). If you do not specify a particular input character on a state, we assume the DFA automatically fails on that input character (ex. The above DFA fails when it receives 'b' ). As in the assignment # 2, \* matches zero or more occurrence of the previous character while + matches one or more occurrences of the previous character. . matches either 'a' or 'b'. ^ is the start of the character string while \$ is the end of character string.

3a)  $^ab^*ba^*$  (7pts)

3b)  $^ab.+ab^*$  (8pts)

4. (Type and Size) What is the output of this program on a lab machine? If some output is undefined, say undefined. (20pts)

```
#include <stdio.h>
#include <string.h>

struct Table {
    int key;
    int value;
};

int main(void)
{
    double i, *p, **pp;
    struct Table t, *pt;
    char strA[] = "abc";
    char strB[] = "a\0bc";
    char strC[] = { 'a', 'b', 'c' };

    p = &i;
    pp = &p;
    pt = &t;

    printf("(1) = %d\n", sizeof(i));
    printf("(2) = %d\n", sizeof(p));
    printf("(3) = %d\n", sizeof(*p));
    printf("(4) = %d\n", sizeof(*pp));
    printf("(5) = %d\n", sizeof(t));
    printf("(6) = %d\n", sizeof(*pt));
    printf("(7) = %d\n", strlen(strA));
    printf("(8) = %d\n", strlen(strB));
    printf("(9) = %d\n", sizeof(strB));
    printf("(10) = %d\n", strlen(strC));
    return 0;
}
```

5. Simple Programs (15pts)

a) What is the output of f(x)? (5pts)

```
void f(char *s)
{
    char *p = s;
    char *q = s;
    while (*s)
        s++;
    for (s--; s > p; s--, p++) {
        char c = *s;
        *s = *p;
        *p = c;
    }
    printf("%s", q);
}
```

```
...
char x[] = "abcde";
f(x);
```

b) What does f(798) produce? (2pts) What does the function do? (3pts)

```
void f(unsigned int n)
{
    do {
        putchar ('0' + (n % 10));
    } while (n /= 10);
    putchar ('\n');
}
```

c) We may rewrite (b)'s code like following. Does this produce the identical output? If not, when is the answer different? (5pts)

```
void f(unsigned int n)
{
    for (; n; n /= 10)
        putchar('0' + (n % 10));
    putchar('\n');
}
```

## Mini class evaluation

Please *detach* this page and submit it separately. Your feedback will be invaluable and will be used to improve the second half of this course. I will greatly appreciate your help! (If you need more space, please use the back of this sheet)

1. In general, I think I learn (1: almost nothing, 2: somewhat less than I hoped, 3: similar amount as other courses, 4: about the right amount that I wanted, 5: a lot more than I expected) about C programming through this course.
2. In regard to the learning amount in (1) , I am (1: completely dissatisfied, 2: somewhat unhappy 3: neutral, 4: satisfied, 5: very satisfied).
3. How are the level of class materials? (pick a number (1-5): 1: too difficult, 5: too easy)
4. How educational are the assignments? (pick a number (1-5), 1: least educational 5: most educational)
5. How useful are the precepts? (pick a number (1-5), 1: least useful, 5: most useful)
6. Overall class evaluation so far (1:very poor, 2: poor, 3: average, 4: good, 5: very good)
7. List the things you like about the course and you want this course to improve on.