Introduction to Computer Science and C Programming-Final

2017/01/11

ID:________________ Name:_________________

pts Total, 30 min

※If you don't understand particular English word, you may ask the teaching assistant to assist you.
※No Discussion. We will ask you to leave and give you zero point if any suspicious behavior is found

1. (5pts) True or False (*Please fill T/F in the blank*)

_____ 1. To read~~s~~ the binary contents of a file *fp* into the array *a*,
you can use "fread(a, sizeof(a[0]), sizeof(a) / sizeof(a[0]), fp);" T

_____ 2. "*int a[3]={1};*" represents the initial value of *a* is {1, 1, 1}. F

_____ 3. Both floating-point numbers and strings can be tested in *switch* statements F

_____ 4. "int *p = (int *) calloc(100 * sizeof(int));" will allocate 100 values of integer type,
and stores a pointer to that array in *p* F

_____ 5. In 2's Complement Notation System, n bits is able to represent decimal $+2^n - 1 \sim -2^n$ F

2. (21pts) **Multiple-choice Questions** *(There may exist one or more than one answer. You will lose ? pts till there is no pts to lose for one wrong choice in each question)*

(1) Which of the following expressions **produce the value 0**? ABDE

ANS:____________________

(A) 10 / 20
(B) ! 5
(C) 0 || 20
(D) 4 % 2
(E) 1 && 0

(2) Which of the follow statements are **correct**? CDE

ANS____________________

(A) -1.375 in 8-bit length floating-point notation is 01011011.
(B) Both *abc123* and *123abc* are valid variable names in C language.
(C) For a variable *i =12345*, the statement "*printf("%3d", i);*" will show *12345* on the screen.
(D) "*while(n) {…}*" executes the loop body infinitely if *n* is a nonzero number (without any break statement in the loop body), including negative numbers.
(E) Given a declaration " *char arr[10]="123\n";* ", then the return value of *strlen(arr)* is 4.

(3) Given an array declaration as following.

```
char *I_LOVE_NCTUCS[] = {"I love C Programming",
                         "I love Physics",
                         "I love Calculas",
                         "I love Linear Algebra",
                         "I love Digital Circuit Lab",
                         "TAs are handsome and beautiful!"};
```

Which of following statements won't cause a program crash and are able to **show exactly "I love C Programming" on the screen**? ACD

ANS:____________________

(A) printf("%s", *I_LOVE_NCTUCS);
(B) puts(I_LOVE_NCTUCS);
(C) fprintf(stdout, "%s", I_LOVE_NCTUCS[0]);
(D) printf("%s", *(&I_LOVE_NCTUCS[0]));
(E) printf("%s", *I_LOVE_NCTUCS[0]);

3. (42pts) Notation Conversion

When changing the notation, the relative position of each number should not be changed. All the number in the expression is 1-digit natural number. The order of infix is just like arithmetic order.
(Prefix notation = Polish notation ; Postfix notation = Reverse polish notation)

Example:
Infix: 3+5 => Postfix: 35+ (53+ is wrong).
Prefix: **123 => Infix: (1*2)*3 or 1*2*3 (just choose one to write, and 1*(2*3) is wrong)

(1) Infix to prefix 3+5*8-7/6+4	(2) Infix to postfix ((1+3)-2*4)/6+5
(3) Prefix to infix -9*+/**876542	(4) Postfix to infix 356789-/**2*+
(5) Prefix to postfix -*+327*4+81	(6) Infix to prefix (1+2)*(3/4+5-6)*7-8/9

ANS:

(1) +-+3*58/764	(2) 13+24*-6/5+
(3) 9-(8*7*6/5+4)*2 Or 9-(((8*7)*6)/5+4)*2	(4) 3+(5*(6*(7/(8-9))))*2
(5) 32+7*481+*-	(6) -**+12-+/34567/89

4. (16pts) Please identify the illegal assignment statements:

```
int main(){
    int a = 1, b = 2;
    int *p = &a;
    const int *q = &b, n = 5;

     A)   p = q;

     B)   *p = 3;

     C)   q = p;

     D)   q = &a;

     E)   *q = 4;

     F)   n = b;

     G)   a = b;

     H)   a = n;


    return 0;
}
```

ANS:

A E F

5. (16 pts) Which of the following statements may crash the program (incorrect memory operations)?

```
#include<stdio.h>
#include<stdlib.h>
int* fun1() {
    int arr[4] = {1, 2, 3};
    return arr;
}

int main() {
    int arr[4] = {1, 2, 3};
    int *p1, *p2, *p3, *p4;

     p1 = fun1();
     p2 = (int*) malloc(sizeof(arr));
     p3 = arr;
     p4 = p2;

     A)   *p1 = 0;

     B)   *p2 = 0;

     C)   *p3 = 0;

     D)   *p4 = 0;

     E)   free(p1);

     F)   free(p2);

     G)   free(p3);

     H)   free(p4);

    return 0;
}
```

ANS:

A E G H