



Final Exam

Term: Fall 2018 (Sep4-Dec4)

Student ID Information

Last name: _____ First name: _____

Student ID #: _____ CS.Dal.Ca userid: _____

		Grade Table	
		Question	Score
Course ID:	CSCI 2132	1	/12
Course Title:	Software Development	2	/12
Instructor:	Vlado Keselj	3	/10
Date of Exam:	6 Dec 2018	4	/7
Time Period:	Start: 15:30 End: 18:30	5	/4
Duration of Exam:	3 hours	6	/10
Exam Location:	Dalplex	7	/8
Number of Exam Pages: (including this cover sheet)	21 pages	8	/9
Exam Type:	Closed Book	9	/8
Additional Materials Allowed:	One letter-format paper (8.5"×11") with anything written or printed on it (both sides). No textbooks, computers, calculators, or other aids are allowed.	10	/11
		11	/8
		12	/10
		13	/9
		14	/13
		15	/10
		16	/10
		17	/12
		Σ	/163

1. (12 points) **True-false questions.** 2 points each. No justification necessary, but it may be helpful if the question seems ambiguous.

a) (2 points) Multics is an advanced version of Unix, developed from Unix version 4.

b) (2 points) The `cut` command is used to display first `n` lines from a text file.

c) (2 points) In a Unix-style system, each process is assigned a memory space, divided into the following four parts: code, stack, heap, and queue.

d) (2 points) The `calloc` C function is used to make a system call to the kernel of the operating system.

e) (2 points) The `gdb` command `break` is used to mark a line in the program where program execution will stop and give us a chance to examine the state of the variables.

f) (2 points) The following C code `char *f="%d\n"; printf(f, 10);` is valid and prints the number 10, followed by a newline character.

2. (12 points) Multiple-choice. Circle the *single* best answer.

a) (3 points) If `s1` and `s2` are two string variables, and `i` is an integer variable; which of the following lines is NOT a valid use of string functions?

- A. `strcmp(i, s1);`
 - B. `strcpy(s2, s1);`
 - C. `strcat(s1, s2+i);`
 - D. `i = strlen(s1);`
-

b) (3 points) Unix has a quite general concept of a file, including several different file types. Which of the following is NOT a file type in Unix?

- A. symbolic link
 - B. job
 - C. socket
 - D. directory
-

c) (3 points) If `p` and `q` are pointers to `int`, pointing to elements of an array, which of the following statements is not valid:

- A. `p = q + 2;`
 - B. `*p = p - q;`
 - C. `*p = p + q;`
 - D. `*p = *p + *q;`
-

d) (3 points) Which of the following lines is a correct way to use the function `scanf`?

- A. `int i=0; scanf("%d", i);`
- B. `int i=0; int *p=&i; scanf("%d", *p);`
- C. `char s[10]="1234567"; scanf("%3s", s);`
- D. `char c; scanf("%c", c);`

3. (10 points) Program Output (pointers).

What is the output of the following program? Include notes or diagrams that justify your answer.

```
int a[10] = {1, 20, 3}, i;  
int *p = &a[3];  
int *q = p+2;  
  
for (i=3; i<10; i++) a[i] = a[i-2]+1;  
  
*(++q) = *(++p); q++; *(q++) = *(p++);  
  
printf(" %d %d %d\n", *a + *q - *p, q-p, *p-*q);  
  
for (i = 4; i < 8; i++) printf(" %d", a[i]);
```

4. (7 points) **Program Output (strings)**. What is the output of the following program? Include notes or diagrams that justify your answer.

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

int main() {
    char s[90];
    char *p = s;
    strcpy(s, "12345-");
    strcat(p, "ABCDE");
    p += 6;
    printf("%d %d %d\n", strlen(s), strlen(p), p-s);
    *--p = '\0'; p+=1;
    printf("(%s) (%s)\n", s, p);
    strcat(s,"XYZ");
    printf("(%s) (%s) (%s)\n", s, p+3);
    return 0;
}
```

Note: The line `printf("(%s) (%s) (%s)\n", s, p+3);` was supposed to be `printf("(%s) (%s) (%s)\n", s, p, p+3);`. The students were instructed to cross out one `(%s)`, or any other reasonable assumption based on demonstrated knowledge of `printf`, `strings`, and `pointers` would be acceptable.

5. (4 points) **Program Output (strcmp).** What is the output of the following program? Include notes or diagrams that justify your answer.

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

void pr(char *s, char *p) {
    int i = strcmp(s,p);
    if ( i == 0 )
        puts("strcmp == 0");
    else if ( i < 0 )
        puts("strcmp < 0");
    else
        puts("strcmp > 0");
}

int main() {
    char s[9] = "abc-78";
    char *p = "abf-192";

    pr(s, p); pr( s + 3, p + 5);
    return 0;
}
```

Note: `pr(s + 3, p + 4);` was supposed to be `pr(s + 4, p + 5);`. The students were instructed to make this change, but using '3' is also fine, the problem is just a bit harder since we need to remember that '-' comes before digits in the ASCII order. Even if student does not remember the order but notices that it is relevant makes the answer correct.

6. (10 points) Single command line. For each of the following questions, write a single Unix command line to perform the task required. You can use pipes and the list of allowed commands are: cut, ls, grep, egrep, sort, uniq, wc

a) (3 points) Print a list of files in the directory `./dir1` that have names starting with an uppercase letter and ending with `.txt`.

b) (3 points) The file `words.txt` contains one word in each line. Print the number of words in this file that are six or seven characters long, start with 'p' or 'b' and end with 'ing'.

b) (4 points) The file `games.txt` contains a list of hockey games in the following format "`team1:team2=score1:score2`" in each line (example: `Jets:Sabres=3:2`). If the first team in each game is the home team, and the second team is the visiting team, write a command to print out all visiting teams, sorted without repetition.

7. (8 points) Give concise answers (permissions, code).

a) (4 points) What permissions are assigned to the file 'f.sh' with the command: `chmod 673 f.sh` ? Can the user (owner) execute the file?

b) (4 points) In the following function, `N` is a compile-time constant:

```
int f(int a[][N]) {  
    printf("%d\n", &a[0][0] - &a[2][6]);  
}
```

If the function prints the number `-20`, what is the value of `N`? Explain.

8. (9 points) Give concise answers (testing, git/svn).

a) (4 points) What is the difference between white and black box testing?

b) (5 points) When we submit changes in 'git' to a remote repository, we use one additional command compared to when doing the same operation in 'svn'. This is related to an essential difference between 'git' and 'svn'. Briefly explain this difference.

9. (8 points) Large program organization.

(8 points) A C program consists of three files `main.c`, `aux.c`, and `btree.c`, and a common header file `b.h`, which is used in all C files. The program executable is called `btree`. Write Makefile rules for the targets: `btree`, `main.o`, `aux.o`, and `btree.o`, for modular compilation and linking of the program.

10. (11 points) Give brief answer.

a) (6 points) Briefly describe the problem of fragmentation of heap. Make sure to describe two types of fragmentation mentioned in class.

b) (5 points) Briefly describe what the `realloc` function does and its time efficiency.

11. (8 points) Brief Answers and Code snippets

a) (4 points) Briefly describe the implicit type conversion in an assignment. Give an example.

b) (4 points) The following function reads some characters from the standard input and stores them in a string on a heap without wasting space. Fill in the missing code and explain how much of input it will store.

```
char *f() {
    int ch, size = 100, count=0;
    char *b = malloc( size );
    while (EOF != (ch = getchar())) {

        if (count > size - 2)

            b = _____ (b, size *= 2); /* Fill in blanks */

        b[count++] = ch;
    }

    b[count] = '\0';

    b = _____ ( b, _____ );
                                     /* Fill in blanks */

    return b;
}
//How much input is read?
```

12. (10 points) Shell scripting

Given the bash script below, briefly explain the lines (1) to (7). Explain the purpose of this script based on your observation. [2pt]

```
if (( $# != 2 )); then                #(1) 2pt
    echo Error
    exit 1
fi

for (( i = $1; $i <= $2; i = $i + 1 )) do    #(2) 2pt
    if [ -f test$i.in ]; then                #(3) 1pt
        ./a.out < test$i.in > test$i.new    #(4) 1pt
        diff test$i.out test$i.new         #(5) 1pt
    fi                                       #(6) 0.5pt
done                                         #(7) 0.5pt
```

13. (9 points) File operations in C

(9 points) Briefly explain the standard function `fopen`. What are the parameters and what the function returns?

14. (13 points) Write a C program.

Assume that you are working on a program to process games of hockey teams and collect their points. The data is organized in a linked list where a node structure is:

```
struct node {
    char *team;          /* team    is the team name */
    int  points;         /* points are accumulated team points */
    struct node *next;  /* next   is pointer to next node */
};
```

(a) (4 points) Write a C function `teamPrint` that takes a pointer to the above node structure and prints a line with the team name, the number of points, and a new line. For multiple lines to be aligned, print team name to at least 12 characters width, and points to at least 2 digits width, with a space between them. An output example is “`Maple_Leafs 4`”. The start of the function is:

```
void teamPrint(struct node *n) {
```

(b) (9 points) Write the C function `teamNew` which allocates a node structure, sets the team name `team` to given name `tname`, sets `points` to 0, and the `next` pointer to NULL. The team name should also be allocated on heap and copied from the given name since the given name may have temporary life span. No error checking is required. The function returns the pointer to the new node. The start of the function is:

```
struct node *teamNew(char *tname) {
```

15. (10 points) Write a C program (continued)

(continuation of the previous question) Write a C function `frontPull` which takes a head pointer of a linked list `head` and a team name `tname` as the arguments. If the linked list contains a node with the team name `tname`, the function moves the corresponding node to the start of the list (head of the list). If the list does not contain a node with such name, a new node is created using the function `teamNew` from the previous page, and the new node is added to the start of the list. The head pointer of the new list is returned. The function prototype is: `struct node* frontPull(struct node* head, char* tname);`

16. (10 points) Write a C program (continued)

(continuation of the previous question) Write a C function `frontPush` which takes a head pointer of a linked list `head` as an argument. Assume that the linked list is sorted **starting from the second element of the list** by **points** in decreasing order (from high to low). The function moves the first element to an appropriate position in the list so that the whole list is sorted by points in the decreasing order. The head pointer of the resulting list is returned. The function prototype is:
`struct node* frontPush(struct node* head);`

17. (12 points) Write a C program (continued)

(*continuation of the prev. question*) Write a C program (just the function `main`) that uses the previous functions to read game scores and print team names with their total points sorted in a decreasing order of points. You can assume that team names have no spaces and are not longer than 99 characters, there are no tie games, and a team gets 2 points for a win and 0 points for a loss. The input format is shown below. Decide when input ends by using the return value of the `scanf` function. The output should use the function `teamPrint` to print output as shown below. If some teams have the same number of points, their relative order is not important (it is random).

<i>Sample Input</i>	<i>Sample Output</i>
Bruins : Maple_Leafs = 2:4	Maple_Leafs 4
Bruins : Panthers = 0:5	Penguins 2
Jets : Penguins = 3:6	Panthers 2
Maple_Leafs : Sabres = 4:3	Sabres 0
	Jets 0
	Bruins 0

An extra page for the last question.

(extra blank page 1 provided)

(extra blank page 2 provided)