

EXAMINATIONS – 2018

TRIMESTER 1

NWEN 241

SYSTEMS PROGRAMMING

Time Allowed: TWO HOURS

CLOSED BOOK

Permitted materials: Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.

No electronic dictionaries are allowed.

Paper foreign to English language dictionaries are allowed.

Instructions: Attempt ALL ELEVEN (11) questions.

There are TWO sections:

- SECTION A - C programming [80 marks]
- SECTION B - Python programming [40 marks]

In SECTION A, there are EIGHT (8) questions:

1. Operators and Operator Precedence. [6 marks]
2. Control Structures. [10 marks]
3. Arrays, Characters and Strings. [8 marks]
4. Arrays and Pointers. [10 marks]
5. Storage Classes. [18 marks]
6. Derived Types and Dynamic Memory. [10 marks]
7. File I/O. [8 marks]
8. Process Management. [10 marks]

In SECTION B, there are THREE (3) questions:

9. Python Fundamentals. [20 marks]
10. Understanding a Python3 Program. [10 marks]
11. Fixing a Python3 Program. [10 marks]

The examination consists of 120 marks in total.

SECTION A C Programming1. Operators and Operator Precedence. **(6 marks)**

(a) What value does the C expression $10 / 4$ evaluate to? **(1 mark)**

2

(b) What is the problem (if any) in the following C statement:? **(2 marks)**

```
int rem = 10.0 / 4 % 2;
```

The right-hand side expression is evaluated as $(10.0/4)\%2 = 2.5\%2$. This is invalid or an error as the % operator requires both operands to be integers.

(c) What is the value of *i*, *j*, and *k* after the last statement in the following C code snippet:? **(3 marks)**

```
int i = 5, j = 10, k = 1;  
(k += 3*--i) - j++;
```

i = 4
j = 11
k = 13

2. Control Structures. (10 marks)

(a) What is the output of the following C program? (2 marks)

```
int main(void)
{
    for (int a = 0; a < 10; a++)
        if (a % 2)
            printf("%d ", a);
    return 0;
}
```

1 3 5 7 9

(b) Rewrite the following code using a for-loop. (2 marks)

```
int main (void)
{
    int j = 5;
    while(j >= 0)
        printf("%d ", --j);
    return 0;
}
```

The for-loop should output 4 3 2 1 0 -1. Possible solutions:

```
int main (void)
{
    int j;
    for(j = 5; j >= 0 /* or j > -1 */; j-- /* or --j */)
        printf("%d ", j-1);
    return 0;
}
```

Or:

```
int main (void)
{
    int j;
    for(j = 4; j >= -1 /* or j > -2 */; j-- /* or --j */)
        printf("%d ", j);
    return 0;
}
```

(c) A student was given the following obfuscated C program: (3 marks)

```
int main(void){int a=2,b=1,n=0,z=3;if(n>0)
if(a>b)z=a;else z=b;printf("%d",z);return 0;}
```

What would be the output from the program? Explain by reformatting the code (using newlines and proper indentation).

The program can be reformatted as follows:

```
int main(void)
{
    int a=2, b=1, n=0, z=3;

    if(n > 0)
        if(a > b)
            z = a;
        else
            z = b;
    printf("%d",z);
    return 0;
}
```

Note that the else clause is associated with the last/closest if clause. Clearly, the output of the program is 3, as it will not even enter the outermost if.

(d) What will be the output of the following program?

(3 marks)

```
int main(void)
{
    int i, j, k = 0;
    for (i = 0; i < 5; i++)
        for (j = 0; j < i; j++) {
            k = i + j - 1;
            printf("%d", k);
            break;
        }
    return 0;
}
```

0 1 2 3

3. Arrays, Characters and Strings. (8 marks)

(a) An array has been declared as: **(2 marks)**

```
long array[] = {1,2,3,4,5, ...};
```

Write a C expression that will give the number of elements of the array.

`sizeof(array)/sizeof(long)`

(b) What would be the output of the following C program? **(2 marks)**

```
int main(void)
{
    char str[] = "NWEN241";
    int sum = 0, i = 0;
    while(str[i])
        if(isdigit(str[i++]))
            sum++;
    printf("%d", sum);
    return 0;
}
```

3

(c) What would be the output of the given C program? Explain. **(4 marks)**

```
void transpose(char *str)
{
    int len = strlen(str);
    for (int i=0; i<len; i++)
        str[i]++;
}

int main(void)
{
    char *string = "QWERTY";
    transpose(string);
    printf("%s", string);
    return 0;
}
```

The program would cause a segmentation fault or some type of run-time error, hence no output would be displayed. This is because string is a pointer to a string literal. The string literal is passed to the transpose() function, which tries to modify it. This causes the error to occur.

4. Arrays and Pointers. **(10 marks)**

(a) A C program has the following declaration:

(4 marks)

```
int numbers[] = { 1, 0, 5, 8, 3, 4, 2, 7, 9 };
```

Answer the following [1 mark each]:

i. What is the value of `numbers[4]`?

3

ii. What is the value of `*numbers + 4`?

5

iii. What is the value of `*(numbers + 4)`?

3

iv. What is the value of `*(numbers + *numbers + 4)`?

4

(b) A C program has the following statements.

(6 marks)

```
short a[] = {1, 2, 4, 8, 16, 32};  
short *pa = a;  
short **ppa = &pa;
```

Suppose each short integer quantity occupies 2 bytes of memory. If the array `a` is at (decimal) address 1608, `pa` is at (decimal) address 1800, and `ppa` is at (decimal) address 1804, then [1 mark each]:

i. What value is represented by `a`?

1608

ii. What value is represented by `ppa`?

1800

iii. What value is represented by `pa + 4`?

$1608 + 4 * 2 = 1616$

iv. What value is represented by `*(pa + 4)`?

$a[4] = 16$

v. What value is represented by `*ppa + 4`?

$pa + 4 = 1616$

vi. What value is represented by `*(ppa + 4)`?

Invalid or unknown

5. Storage Classes. (18 marks)

In the following program, there are variables A, B1, B2 and C.

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

int A; // For subquestion a).

void local_function (int B_in) {

    int B1;           // For subquestion b).
    static float *B2; // For subquestion c).

    B1 = ++A + B_in;
    printf("%d %d\n", B_in, B1); // For subquestion e).

    //...

    B2 = (float *) calloc (4, sizeof(float));

    //...

    free(B2);
}

int main()
{
    int C = 0; // For subquestion d).

    A = C++;
    local_function (A);
    local_function (++C);

    return 0;
}
```

Answer the following FIVE (5) subquestions:

- (a) In the above program, what is the storage class of variable A? Which segment (i.e. Data, Stack or Heap) is A allocated memory space? **(4 marks)**

Extern or external storage class. It is stored in the data segment.

- (b) In the above program, which segment is B1 allocated memory space? What is the storage duration of B1? **(4 marks)**

Auto or automatic storage class. It's storage duration is local, or within local_funtion().

- (c) In the above program, in which segment is B2 allocated memory space? B2 points to a block of dynamically allocated memory; in which segment is this block of (dynamic) memory allocated space? **(4 marks)**

B2 itself is a static variable, hence it is allocated in the data segment. B2 points to a block of memory that is allocated in the heap segment.

- (d) In the main function, there is a variable c. What is the storage class of variable c and where is it visible? **(4 marks)**

Auto or automatic storage class. It is visible within the main() function.

- (e) What is the output of the above program? **(2 marks)**

0 1
2 4

6. Derived Types & Dynamic Memory. (10 marks)

- (a) Define a structure that can represent the dimensions of a rectangle, with tag `rect` and consisting of 2 float members `width` and `length`. (2 marks)

```
struct rect {
    float width;
    float length;
};
```

Or:

```
struct rect {
    float width, length;
};
```

- (b) Use typedef to define a new type `rect_t` from the structure defined in (a). (2 marks)

```
typedef struct rect rect_t;
```

- (c) Consider the following declaration where `rect_t` is the type defined in (b): (2 marks)

```
rect_t *p;
```

Write a C statement that will allocate an array of 20 `rect_t` elements, and let `p` point to that memory.

```
p = (rect_t *) calloc(20, sizeof(rect_t));
```

Or:

```
p = (rect_t *) malloc(20 * sizeof(rect_t));
```

- (d) Write a function with prototype (4 marks)

```
rect_t *create_rect(float w, float l);
```

that will allocate memory for a `rect_t`, set the width and length to `w` and `l`, respectively, and return a pointer to the allocated memory.

```
rect_t *create_rect(float w, float l)
{
    rect_t *p = (rect_t *) malloc (sizeof(rect_t));
    p->width = w;
    p->length = l;

    return p;
}
```

7. File I/O. (8 marks)

- (a) Write a C statement that will open a binary file `bfile.bin` for writing such that writes will occur after the end of file. Let `fp`, a variable of type `FILE` pointer, point to the opened file stream. (2 marks)

```
FILE *fp = fopen("bfile.bin", "ab");
```

Or:

```
fp = fopen("bfile.bin", "ab");
```

- (b) Suppose that `infp` is a `FILE` pointer that points to an opened binary file stream for reading, write a C statement that will reposition the stream to a location that is 100 bytes from the end of file. (2 marks)

```
fseek(infp, -100, SEEK_END);
```

- (c) Consider the following C code snippet: (4 marks)

```
char c;
FILE *infp = fopen("infile.txt", "r");
FILE *outfp = fopen("outfile.txt", "w");
while( (c=getc(infp)) != EOF ) {
    putc(++c, outfp);
}
fclose(infp);
fclose(outfp);
```

If the contents of `infile.txt` is

Bnlotsdq

What would be the contents of `outfile.txt`?

Computer

8. Process Management. (10 marks)

You are given the following C program.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int gvar = 10;

int main(void)
{
    int lvar = 20;
    pid_t pid;

    printf("fork test\n");
    if ((pid = fork()) < 0) {
        printf("fork error\n");
    } else if (pid == 0) { /* child */
        gvar++;
        lvar++;
    } else { /* parent */
        wait(NULL);
    }

    printf("%ld %d %d\n", (long)getpid(), gvar, lvar);
    exit(0);
}
```

Answer the following THREE (3) subquestions.

- (a) Assume that the fork is successful and that the parent process ID is 16231 while the child process ID is 16232. What is the output of the program?

(3 marks)

```
fork test
16232 11 21
16231 10 20
```

- (b) Explain how the `fork()` function executes and how/when the variables `gvar` and `lvar` are changed. **(5 marks)**

Upon invocation, the `fork()` function creates another process using the same image as the current process. In the parent process, the return value of `fork()` is the child process id while in the child process, its return value is 0. The variables `gvar` and `lvar` are only changed in the child process.

(c) Explain how the `wait()` function works. **(2 marks)**

The `wait()` function causes the parent process to wait for its child process to exit/terminate before continuing execution.

SECTION B Python Programming**9. Python Fundamentals (20 marks)**

- (a) Identify the types of the variables `my_var1` and `my_var2` in the following code snippet: **(2 marks)**

```
my_var1 = 3.1415
my_var2 = '3.1415'
```

`my_var1` is float, `my_var2` is string (or str)

- (b) Briefly explain what the built-in function `input()` does and write a short piece of code to illustrate its use. **(4 marks)**

The `input()` function waits for keyboard input from user and when enter is pressed, returns whatever the user has typed. Example:

```
name = input("Please enter your name:")
```

- (c) Briefly describe the dictionary and tuple data collections. Give an example defining a dictionary and a tuple in python3. **(4 marks)**

A dictionary stores key-value pairs while a tuple is an immutable list. Example of a dictionary definition:

```
my_dict = {'red': 1, 'green': 2, 'blue': 3}
```

Example of a tuple definition:

```
my_tuple = (1, 2, 3)
```

- (d) Write the output when the following code is executed. **(4 marks)**

```
my_list = ['a', 2, 3, 4, "e", 6, 7, 8, 9, "k"]
print(my_list[-4:])
```

7, 8, 9, "k"

- (e) Consider the following python3 function: **(4 marks)**

```
def area(length, width):
    return length*width
```

Write two statements that are examples of calling the function, one using positional arguments and the other using keyword arguments.

Using positional arguments:

```
a = area(1, 2)
```

Using keyword arguments:

```
a = area(length = 1, width = 2)
```

Or:

```
a = area(width = 2, length = 1)
```

- (f) Briefly describe the effect of executing the following code snippet: **(2 marks)**

```
file_handle = open("testfile.txt", "a+")
```

If "testfile.txt" does not exist, it is created and opened in read-write mode. If it exists, it is opened in read-write mode. Writes will occur after the end of file.

10. Understanding a Python3 Program. (10 marks)

Consider the following python3 function:

```
def strval(n=1, w="aaa"):
    """ w must contain at least 3 characters
    """

    if w[2] == 'u':
        return n*2
    elif n > 5:
        return (w+w)
    elif w == w[::-1]:
        return (1 == 2)
    return w[0]
```

Fill in the table below. In each column, write the return value and type for each of the following function calls:

Function Call	Return Value	Return Type
strval()	False	Boolean or bool
strval(3,"Kaukau")	6	Integer or int
strval(6,"Peka")	PekaPeka	String or str
strval(9,"Mount Victoria"[-3:])	riaria	String or str
strval(w="Rimutaka")	R	String or str

Student ID:

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

11. Fixing a Python3 program. (10 marks)

You were asked to write a python3 program for counting the occurrences of a given word from any text file, and the program should output a line:

```
word    num_occurrences
```

where `num_occurrences` is the number of times `word` appeared in the entire text file. The space between `word` and `num_occurrences` is a single tab. As an example, if the given word is `cat` and it appears 12 times in the text file, then the output of the program should be:

```
cat    12
```

Fortunately, a friend of yours told you that he had already written the program and sent you a copy of the source code. Unfortunately, when you ran his program, it didn't work.

You asked another friend to help you debug the program. Upon examination, she told you that the program has at least FIVE (5) errors. The errors can be found in the 8-line function `count_words`, shown below:

```
1 def count_words(filename, word):
2     count = 1
3     f = open(filename)
4     for w in read(f).split():
5         if w is word:
6             count++
7     print("{0}\t{1}".format(word, count))
8     close(f)
```

In the space provided below, identify and correct each of these errors, in the following manner:

1. state the line number;
2. clearly explain the error; and
3. provide the correct code to replace the erroneous one.

Note: Ignore indentation errors and there may be zero or more errors per line, i.e. not every line has errors.

Errors:

Line 2: the variable `count` is initialized with incorrect value

Correct statement: `count = 0`

Line 4: `read()` is used incorrectly – it is a method on file handle

Correct statement: `f.read().split()`

Line 5: the use of `'is'` will not work correctly

Correct statement: `if w == word:`

Line 6: Python has no increment operator

Correct statement: `count += 1` or `count = count + 1`

Line 8: `close()` is used incorrectly, it is a method on file handle

Correct statement: `f.close()`

Student ID:

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Student ID:

* * * * *