School of Computer Science & Software Engineering
The University of Western Australia

Mid-Semester Test
September 2017

Systems Programming 2002 (CITS2002)

This paper contains 1 section

This paper contains: 8 pages (including this title page)
Time allowed: 45 minutes (no additional reading time)

Each question is worth 1 mark.
Marks for this paper total 20.
Candidates should attempt ALL questions.
This is a closed book examination – no written materials or calculators are permitted.

PLEASE NOTE

Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.

Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.

1.

(1) Which of the following statements is/are true about a C99 compiler?

i. A C99 compiler will only (re-)compile programs that have changed.
ii. A C99 compiler will detect all errors within a C99 program.
iii. A C99 compiler will generate a form of machine-code that can be understood across all different machine architectures.

A. None of i., ii., and iii.

B. i. only.

C. ii. only.

D. iii. only.

(2) Which of the following is NOT a valid C99 preprocessor directive?

A. #if

B. #else

C. #import

D. #define

(3) Which of the following words is NOT a valid C99 keyword?

A. for

B. nested

C. break

D. do

(4) Consider an integer variable `x` that contains a number with at least two digits. Which of the following correctly returns the TENS digit (the second digit from the right) from the variable?

A. x / 100 % 10

B. x % 100 / 10

C. x % 10 / 10

D. x / 10 % 100

(5) The following function correctly converts a lowercase ASCII character into an uppercase ASCII character:

```
char upperCase(char c)
{
  return (c + 'A' - 'a');
}
```

What properties about characters are essential in order for this function to behave correctly?

i. The language provides mechanisms to convert between character constants and ASCII values, thus allowing numerical operations on characters.
ii. ASCII values are contiguous such that successive characters have successive ASCII values.
iii. Uppercase characters have an ASCII value greater than their corresponding lowercase character.

A. i only.

B. i and ii only.

C. i and iii only.

D. None of them.

(6) Which of the following code segments does NOT correctly check if the character variable `ch` holds an uppercase alphabetic character?

A. `'A' <= ch <= 'Z'`

B. `isalpha(ch) && ch <= 'Z'`

C. `isupper(ch)`

D. `isalpha(ch) && !islower(ch)`

(7) Consider the following program stub:

```
int a, b;

void go(int c)
{
  int d;

  << HERE >>
}
```

At the point marked `<< HERE >>` in the code, which variables are in scope?

A. Variable `d` only.

B. Variables `a` and `b` only.

C. Variables `c` and `d` only.

D. All of the variables.

(8) Consider the following function:

```
void go(void)
{
  for(int i = 0; i <= 10; i++)
  {
    if((i % 2) == 1 || (i % 3) == 1)
    {
      continue;
    }
    printf("%i  ", i);
  }
  printf("\n");
}
```

What is printed when the `go` function is executed?

A. `0 1`

B. `0 2 6 8`

C. `0 5 10`

D. `6`

(9) Consider the following functions:

```
void f1(int x)
{
  while(x < 5)
  {
    printf("hello\n");
    x++;
  }
}

void f2(int x)
{
  do
  {
    printf("hello\n");
    x++;
  } while(x < 5);
}
```

Assuming that the same value is passed to both functions, under what circumstances do `f1` and `f2` produce the same output?

A. Only when the value passed is < 5.

B. Only when the value passed is ≤ 5.

C. Only when the value passed is $= 5$.

D. Never, regardless of the value passed.

(10) Which one of the following statements about function parameters is true?

A. Empty parameter lists are declared with the keyword `void`.

B. If there is only one parameter, the parameter list is not required.

C. A function's parameters are known as "actual parameters".

D. A local variable may have the same name as a function's parameter, overriding the use of the parameter.

(11) Which of the following statements about character arrays in C is/are true?

i. Indexing starts from 1.
ii. Character arrays are always terminated by the null-byte.
iii. Array elements are all of the same data type.

A. i. only.

B. ii. only.

C. iii. only.

D. None of i., ii., or iii.

(12) Consider the following two functions:

```
void increment(int i)
{
  i++;
}

void go(void)
{
  int i = 0;
  int sum = 0;

  for (i = 0; i < 10; increment(i))
  {
    sum = sum + i;
  }
  printf("Final value of i = %i\n", i);
}
```

What is printed when the `go` function is executed?

A. `Final value of i = 0`

B. `Final value of i = 9`

C. `Final value of i = 10`

D. None of the above.

(13) Consider the following function:

```
void go(void)
{
  char str[] = "hello world";
  str[5] = '\n';
  str[8] = '\0';
  printf("length = %i\n", strlen(str));
}
```

What is printed when the `go` function is executed?

A. `length = 5`
B. `length = 8`
C. `length = 11`
D. `length = 12`

(14) Which of the following statements about how lines are terminated in text files is true?

A. The end of a line is represented by the null-byte character.
B. The end of a line is represented by the newline character.
C. The end of a line is represented by the carriage-return character followed by the end-of-line character.
D. The way the end of a line is represented is operating-system dependent.

(15) Consider the following code:

```
char buffer[100];
fgets(buffer, sizeof buffer, fp);
```

Which of the following statements about the call to the `fgets` function is always true?

A. `fgets` reads all the bytes in the file up to the next newline character.
B. `fgets` only returns NULL if the end-of-the-file is reached.
C. `fgets` places a null-byte in the character array if reading was successful.
D. `fgets` stops reading when it encounters the special end-of-file character, EOF.

(16) When a process's actions result in it going through the state transition Running → Blocked, which information about the process is LEAST likely to be stored by the operating system:

A. the process's next process state.
B. the process's next instruction to be executed.
C. references to the memory owned by the process.
D. the total execution time (so far) of the process.

(17) An "interrupt service routine" is:

A. an instruction sequence executed by the operating system to service an interrupt.

B. a recovery procedure an operating system executes when a computer crashes.

C. a kind of computer code that interrupts users when they misuse a computer.

D. none of the above three options.

(18) Suppose a disk interrupt occurs when the CPU is executing a process. The next instruction that the CPU executes is:

A. the first instruction of the process waiting for the interrupt.

B. the first instruction of the CPU scheduler.

C. the first instruction of the disk interrupt service routine.

D. the first instruction to terminate the currently running process.

(19) Which one of the following will generally NOT result in a process's termination:

A. an attempt to execute an invalid instruction.

B. an attempt to write data to a full disk-drive.

C. a per-process time limit has been reached.

D. an attempt to access memory not owned by the process.

(20) Why do C libraries and the operating system kernel both check the validity of parameters to system calls?

A. user-written checks are more likely to be flawed than those of the kernel.

B. it is possible to bypass the libraries' checks and call the system calls directly.

C. checking parameters twice avoids any race-condition which could occur between checks.

D. in practice, the kernel doesn't need to check system call parameters if the checks performed by the library succeed.