CS Basics
Fall 2021

Sample Exam – Jan.2020

Name: _____

Instructions: (points) Bitte geben Sie knappe, präzise aber vollständige Antworten. Wenn nicht nach einer Begründung gefragt wurde, sollten Sie auch keine angeben!

Give short, precise and complete answers. If you are not asked for an explanation, you should not give one!

Donnez des réponses courtes, précises et complètes. Si on ne vous demande pas d'explication, n'en donnez pas!

(1 pts) **1.** What is the difference between a `size_t` and an `unsigned int` in C?

(5 pts) **2.** What is the return value of the following program?

```
int foo(int arr[4])
{
  int ret = 0;
  for (int i=0;i<40;i++)
    ret += arr[i,1];
  return ret;
}

int main()
{
  int data[4] = {0,1,2,3};
  return foo (data + 1);
}
```

(5 pts) **3.** What is wrong with the following code? What is the easiest way to fix the issue(s)? (It is ok for you to assume that there are no syntactic errors in this example).

```
enum Rights {
    RIGHT_TO_READ,
    RIGHT_TO_WRITE,
    RIGHT_TO_EXECUTE,
};
struct MyFile {
  enum Rights rights;
  // ...
};
int do_read(struct MyFile f)
{
  if (f.rights & RIGHT_TO_READ) { ... }
```

```
  else abort();
}
int do_write(struct MyFile f)
{
  if (0 != (f.rights & RIGHT_TO_WRITE)) { ... }
  else abort();
}
int do_execute(struct MyFile f)
{
  if (RIGHT_TO_EXECUTE != f.rights & RIGHT_TO_EXECUTE) { ... }
  else abort();
}
int allow_all(struct MyFile f)
{
  f.rights = RIGHT_TO_READ | RIGHT_TO_WRITE | RIGHT_TO_EXECUTE;
}
```

(5^{pts}) **4.** What are all possible outputs of the following program? Assume the `pthread_create()` call does not fail.

```
static int r;

static void *add(void *p) {
  r++;
}

int main(int argc, char ** argv) {
  pthread_t p1;
  pthread_t p2;

  pthread_create (&p1, NULL, &add, NULL);
  pthread_create (&p2, NULL, &add, NULL);
  printf ("%d ", r);
  pthread_join (p1, NULL);
  pthread_join (p2, NULL);
  printf ("%d\n", r);
  return 0;
}
```

(1pts) **5.** Explain the meaning of the `const` keyword. Example: `void func(const void *p);`

(3pts) **6.** What is the return value of the following program? Justify your answer.

```
static int i;
int foo(int i)
{
  int ret = i;
  for (int i;i<4;i++)
    ret += i;
  return ret;
}

int main()
{
  return i+foo(4);
}
```

(3pts) **7.** Which of the following lines are syntactically valid variable declarations in C (as discussed in class and as allowed by the GNU C compiler)?

You will gain $\frac{1}{3}$ point for correct answers and loose a $\frac{1}{3}$ point for incorrect answers! You can choose not to give an answer (which will neither earn nor loose you any points).

(a) int a[];

(b) int a *;

(c) int a[4];

(d) int []a;

(e) int []a[4];

(f) int * a;

(g) int [4]a;

(h) int * a[4];

(i) int **** a;

(1pts) **8.** What are all possible outputs of the following program?

```
int t() {
  printf ("t ");
  return 1;
}
int f() {
  printf ("f ");
```

```
  return 0;
}
int main(int argc, char ** argv) {
  printf("%d %d\n", t() & f(), f() && t());
}
```

(1pts) **9.** What is the output of the following program?

```
int main(int argc, char ** argv) {
  printf("%1.0f %3.0f\n", 10.0, 0.0);
}
```

(3pts) **10.** What is the output of the following program? Explain why.

```
#define S(X) #X
int main(int argc, char ** argv) {
  printf("%c\n",
         *S(5));
}
```

(1pts) **11.** What is wrong with the following C code?

```
#include <stdio.h>

int foo (int v)
{
  for (int j=0;j<v;j++)
    bar(v-1);
}
int bar (int v)
{
  for (int j=0;j<v;j++)
    foo(v-1);
  printf ("%d", v);
}
int main(int argc, char** argv) {
  foo (5);
  return 0;
}
```

(2 pts) **12.** Correct all bugs in the following code snippets (each example can have multiple or no bugs). Sometimes, there maybe multiple ways to fix the problems (since you do not have enough context). In that case, pick one of the appropriate solutions. Making unnecessary changes or adding new bugs will result in points being taken off.

(a)
```
int u;
sscanf("42", "%u", &u);
printf("%u", u);
```

(b)
```
int i = 42;
printf("The answer is %i\n", i);
```

(2 pts) **13.** What is a phony target in a Makefile? Give a simple example.

(1 pts) **14.** What option must be given to gcc to generate debug symbols?

(10 pts) **15.** Implement a function "words()" that takes a string and returns the number of words in the sentence (words are separated by comma, semicolon, space or period). For example, given "Hello, my dear students." the correct return value is 4. Given "CS-basics is awesome!" the correct return value is 3 (because hyphens to not split words).

(10^{pts}) **16.** Complete the following ASM-code. You have to write the procedure **procA** for transforming a number in RAX into a string containing its binary representation (in reverse order). This procedure takes as input the following two registers RSI and RAX. RSI contains the address of an output string. RAX contains a number. The output of the procedure is RDX that contains the length of the produced string. As a side effect, the number in RAX is written as a string at the address that was in RSI.

```
SECTION .bss ; Section containing uninitialized data
BUFFLEN equ 32
Buff:  resb BUFFLEN ; Text buffer itself
SECTION .data ; Section containing initialised data
table db "01"
SECTION .text ; Section containing code
global  _start ; Linker needs this to find the entry point!

;;; Procedure for writing a number in Binary
;;; Output is written in reverse form (lowest bits first)
;;; Input: In RSI the address of the buffer to write
;;;        In RAX the number to read
;;; Output: in RDX the length of the produced string
;;; Side effect: In memory, the number in RAX is written as a string
;;;              at the address that was in RSI.
procA:









ret ; Return the procedure

_start:
mov rax, 20
mov rsi, Buff
call procA
mov rax,1       ; Code for Sys_write call
mov rdi, 1     ; Specify File Descriptor 1: Standard Output
mov rsi, Buff   ; Pass offset of the message
syscall ; Make kernel call
;;; WRITES : 00101 on stdout (since 20 = 10100B)
mov rax, 60 ; Code for exit
mov rdi, 0  ; Return a code of zero
syscall ; Make kernel call
```

(2^{pts}) **17.**

We have the folloging program and we want to know what it does. You must understand this program and write for each register when asked the value it contains.

The value of the register can be presented as a decimal number or an hexadecimal number (in this case, with the prefix 0x).

If a register contains an adress, you will write a number similar to: `0x600711`.

```
SECTION .bss ; Section containing uninitialized data
BUFFLEN equ 16 ; We read the file 16 bytes at a time
Buff:  resb BUFFLEN ; Text buffer itself
SECTION .data ; Section containing initialised data
str1 db "Hello WorLD"
STRLEN equ $-str1
SECTION .text ; Section containing code
global  _start ; Linker needs this to find the entry point!
_start:
nop ; This no-op keeps gdb happy...
mov RSI, str1
mov RBX, STRLEN
mov RCX, 0
mov RDX, 0
.loop:
mov AL, [RSI]
cmp RAX,'a'
jb .cl
jmp .next
.cl:
inc RDX
.next:
inc RSI
inc RCX
dec RBX
jnz .loop
;;; Write the value of the following registers
;;;
;;; RAX =
;;;
;;; RBX =
;;;
;;; RCX =
;;;
;;; RDX =
;;;
mov rax, 60 ; Code for exit
mov rdi, 0  ; Return a code of zero
syscall ; Make kernel call
```