ECE264 Fall 2021 Exam 1

6:30-7:30PM, September 30

Please keep the answer sheet clean. Do not use the answer sheet as your scratch space. The answer sheet should contain **only** the answers.

Do not write anything that is not the answers. Otherwise, you may lose points.

Please use $\boxed{\textbf{DARK}}$ ink. If your pen is too light, your answer may not be graded.

When you take this exam, you agree with the Purdue Honor Pledge: **advancing a culture of academic integrity, seeing academic dishonesty and cheating as threats to Purdues reputation**.

Value	Character	Value	Character	Value	Character
48	0	65	A	97	a
49	1	66	B	98	b
50	2	67	C	99	c
51	3	68	D	100	d
52	4	69	E	101	e
53	5	70	F	102	f
54	6	71	G	103	g
55	7	72	H	104	h
56	8	73	I	105	i
57	9	74	J	106	j
		75	K	107	k
		76	L	108	l
		77	M	109	m
		78	N	110	n
		79	O	111	o
		80	P	112	p
		81	Q	113	q
		82	R	114	r
		83	S	115	s
		84	T	116	t
		85	U	117	u
		86	V	118	v
		87	W	119	w
		88	X	120	x
		89	Y	121	y
		90	Z	122	z

1 Pointers and Arrays

Consider the following piece of C code:

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

int main(int argc, char * argv[])
{
    int * x = malloc(5 * sizeof(int));
    int * y = malloc(5 * sizeof(int));
    int * * p = &x;
    int * * q = &y;

    for (int i = 0; i < 5; i++) {
        x[i] = i;     //x has values [0, 1, 2, 3, 4]
        y[i] = 4 - i; //y has values [4, 3, 2, 1, 0]
    }

    printf("* * p = %d\n", * * p); // <=== ANSWER A

    printf("* (*q + 1) = %d\n", * (*q + 1)); // <=== ANSWER B

    * q = (*p + 1);
    printf("y[1] = %d\n", y[1]); // <=== ANSWER C

    * p = (*q + 1);
    printf("x[1] = %d\n", x[1]); // <=== ANSWER D

    return EXIT_SUCCESS;
}
```

Each print statement prints out an integer value. The answers to this question are those integer values

Hint: The output of the program will be

```
* * p = [ANSWER A]
* (*q + 1) = [ANSWER B]
y[1] = [ANSWER C]
x[1] = [ANSWER D]
```

Answer:

```
* * p = 0
* (*q + 1) = 3
y[1] = 2
x[1] = 3
```

2 Structures

Consider the following structure definitions:

```
typedef struct {
    int p[10];
    int q[10];
} A;

typedef struct {
    int r[10];
    int s[10];
} B;

typedef struct {
  int x;
  A * y;
  B * z;
} C;
```

ANSWER A:
Assume you have a structure variable `var`:
`C var;`
How would you allocate space for the `y` field of `var`? Your answer should be one line of C code.

ANSWER B:
Assume that all of the fields of `var` are correctly allocated and initialized.
Which of these statements makes the 3rd element of the second field of the `z` field of `var` equal to 7? Your answer should be the number of the correct statement from the list below.

1. `var.z.s[3] = 7;`
2. `var.z->s[2] = 7;`
3. `var->z.s[3] = 7;`
4. `var->z.s[2] = 7;`
5. `var.z.s[2] = 7;`
6. `var.z->s[3] = 7;`

Answer:

ANSWER A: Acceptable answers:
var.y = malloc(sizeof(A))
var.y = (A *) malloc(sizeof(A))
var.y = malloc(sizeof(* var.y))
var.y = (A *) malloc(sizeof(* var.y))
ANSWER B:
```
var.z->s[2] = 7;
```

3 Makefile

Consider the following `Makefile` and answer the questions A - D.

```
1  CFLAGS = -std=c99 -g -Wall -Wshadow --pedantic -Wvla -Werror
2  GCC = gcc $(CFLAGS)
3  EXEC = q3prog
4  TESTFLAGS = -DTEST_COMPARE
5
6  SRCS = add.c compare.c main.c
7  OBJS = $(SRCS:%.c=%.o)
8  # <--- Question A: What is the value of OBJS? --->
9
10 all: $(EXEC)
11
12 # link .o to executable
13 $(EXEC): $(OBJS)
14         # <--- Question B: Fix the following --->
15         $(GCC) $(TESTFLAGS) $(OBJS)    [ANSWER B]    $(EXEC)
16
17 # convert .c to .o
18 %.o: %.c
19         $(GCC) $(TESTFLAGS) -c $<
20
21 testall: test1 test2 test3
22
23 test1: $(EXEC)
24         ./$(EXEC) inputs/input1 > output1
25
26 test2: $(EXEC)
27         ./$(EXEC) inputs/input2 > output2
28
29 test3: $(EXEC)
30         ./$(EXEC) inputs/input3 > output3
31
32 clean: # remove all machine generated files
33         rm -f $(EXEC) *.o *.out *gcda *gcno *gcov
```

ANSWER A: Write down the value of `OBJS`.

ANSWER B: Fix the command at line 15. Your answer should be what needs to go in `[ANSWER B]`

ANSWER C: Using the given `Makefile`, what is the Unix command to test with all test cases (input1, input2, input3)? Your answer should be a single command (plus any needed arguments) you can type at the command line.

ANSWER D: Suppose we have the following in `compare.c`:

```
# ifdef TEST_COMPARE
void func1(void) {
    ...
}
# endif

# ifdef TEST_FUNC
void func2(void) {
    ...
}
# endif
```

When we use the given `Makefile` to test the program, which function(s) is/are compiled and tested? Your answer should be the number of the correct response.

1. `func1`
2. `func2`
3. Both `func1` and `func2`
4. None of them

Answer:

```
A: add.o compare.o main.o
B: -o
C: make testall
D: func1
```

4 Memory Leak

Consider the following function:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#pragma pack(1) // tell compiler not to pad any space
// assume sizeof(char) = 1, sizeof(a pointer) = 8

int main(int argc, char ** argv)
{
    if (argc < 2) {
        printf("Not enough inputs\n");
        return EXIT_FAILURE;
    }

    char ** arr = malloc((argc-1) * sizeof(char *));
    int i = 0;
    for(; i < argc-1; i++) {
        // strlen(argv[i+1]) computes the length of the
        // string argv[i+1], not including the
        // terminating null character, '\0'.
        arr[i] = malloc((strlen(argv[i+1]) + 1) * sizeof(char));

        // strcpy copies the string pointed to
        // by argv[i+1] to arr[i]
        strcpy(arr[i], argv[i+1]);
        printf("%s\n", arr[i]);
    }

    for(i = 0; i < argc-1; i++) {
        free(arr[i]); // release memory
    }
    // <--- Question A: Fix the following code to release memory --->
    free(   [Question A]   );

    return EXIT_SUCCESS;
}
```

Suppose the executable program is named `q4prog`.
We run the program with

```
./q4prog ece 264 midterm1
```

Answer the questions A - C.

Hint: The output of the program will be

```
ece
264
midterm1
```

ANSWER A: Fix the code at line 32 to properly release memory. Your answer should the expression that is the argument to `free`.

ANSWER B: If we do not release memory at line 32, how many bytes of memory will be leaked?

ANSWER C: If we do not release memory at line 29, but we *do* release the memory at line 32, how many bytes of memory will be leaked?

Answer:

```
A: arr
B: 24
C: 17
```

5 Data Types

Consider the following program that reads an input file as integers or floating point numbers.

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

int main(int argc, char * * argv)
{
  if (argc != 2)
    {
      fprintf(stderr, "Need one input file\n");
      return EXIT_FAILURE;
    }
  FILE * fptr = fopen(argv[1], "r");
  if (fptr == NULL) // fopen fail
    {
      return EXIT_FAILURE;
    }
  int count = 0;
  // -------------------
  // treat the input as integers
  count = 0;
  fseek(fptr, 0, SEEK_SET); // return to the beginning of the file
  int ival;
  while (fscanf(fptr, "%d", & ival) == 1)
    {
      count ++;
    }
  printf("The file has %d integers.\n", count); // <=== ANSWER A
  printf("The last read integer is %d.\n", ival); // <=== ANSWER B
  // -------------------
  // treat the input as float
  count = 0;
  fseek(fptr, 0, SEEK_SET); // return to the beginning of the file
  float fval;
  while (fscanf(fptr, "%f", & fval) == 1)
    {
      count ++;
    }
  printf("The file has %d float.\n", count); // <=== ANSWER C
  printf("The last floating-point is %f.\n", fval); // <=== ANSWER D
  // -------------------
  // understand the sizes of different types
  // assume
  // sizeof(char)   is 1
  // sizeof(int)    is 4
  // sizeof(int*)   is 8
  // sizeof(double) is 8
  double darr[] = {2.64, 20.11, 0.9, 2.8};
  int sizediff = (int)sizeof(darr) - (int)sizeof(darr[0]);
  printf("%d\n", sizediff); // <=== ANSWER E
  return EXIT_SUCCESS;
}
```

This is the input file:

```
2 64 20.11 9 17
```

The following table shows bytes of this input file.

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Character	2		6	4		2	0	.	1	1		9		1	7
ASCII (decimal)	50	32	54	52	32	50	48	46	49	49	32	57	32	49	55

Suppose you run the program with the above input file. Please write down the program's output from the five print statements. **Your answers should be ONLY the numbers.**
Answer:
3
20
5
17
24