Family Name:. . . . . **Other Names:** . . . .

Student ID:. . . . . **Signature**. . . . .

NWEN 241: Final Exam

2019, Dec 24 ** WITH SOLUTIONS **

Instructions

- Time allowed: **180 minutes**
- Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.
- Write your answers in this exam paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- You may use dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions		**Marks**	
1.	C Fundamentals.	[10]	
2.	User-Defined Types	[14]	
3.	Arrays and Pointers	[21]	
4.	Dynamic Memory Allocation	[10]	
5.	Data Structures	[16]	
6.	File I/O	[7]	
7.	Low-Level and Socket Programming	[12]	
8.	Process Management	[10]	
		TOTAL:	

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.

Question 1. C Fundamentals **[10 marks]**

(a) **[2 marks]** What will be the output of the following code? Briefly explain how you arrived at your answer.

```
#include <stdio.h>
void main()
{
    int a = 012 , b = 12;
    printf("%d %d", a, b);
}
```

```
10 12 (1 mark)
012 is an octal number equivalent to 10 as an integer and b = 12 is
(1 mark)
```

(b) **[2 marks]** Define a constant PI with value 3.14 using an appropriate preprocessor directive.

```
#define PI 3.14
```

(c) **[2 marks]** What is the output of the following program? A float is 4 bytes.

```
#include <stdio.h>
int main()
{
    float a[15];
    printf("%d", sizeof(a));
   return 0;
}
```

```
Output:  60 (1 mark)
```

(Question 1 continued on next page)

(Question 1 continued)

(d) **[2 marks]** What is the difference between a character array and a string? Show how the word Nihao is stored as a character array and as a string.

```
 A string is an array of characters with the null character /0 after
the last character.  (1 mark)
 char array:  |N|i|h|a|o| | (1/2 mark)
 char array:  |N|i|h|a|o|/0| | (1/2 mark)
```

(e) **[2 marks]** What is the output of the following C program?

```
#include <stdio.h>

int macro_be(int a, int b)
{
    return a*++b;
}

int main(void)
{
    int i = 7; int j = macro_be(1+2, i);
    printf("%d, %d", i, j);
    return 0;
}
```

```
 7, 24 (2 marks)
```

Question 2. User-defined Data Types **[14 marks]**

(a) **[3 marks]** What is the data type of a? Please explain your answer.

```
typedef int numbers;
numbers a = 1;
```

```
a is of type int
numbers is defined to be equivqalent to int by the first line
Alternatively
a is of type numbers, which defined to be equivalent to int
```

(b) **[3 marks]** Define an enumeration type with identifiers `March`, `April`, and `May` having values of 3, 4, and 5, respectively. Use `month` as name of the enumeration type.

```
enum month { March = 3, April, May };
```

(c) **[3 marks]** Consider the following user defined data type:

```
struct student
{
    int id_number;
    char name[20];
    int age;
    char college_name [40];
    int std;
    chr div;
}
```

Write a statement that declares three variables S1, S2, and S3 to be of the student data type.

```
struct student S1, S2, S3 (3 marks)
```

(Question 2 continued on next page)

(Question 2 continued)

(d) **[5 marks]** Based on the data type declaration made above (part c), write code to initialize the S1 variable to contain the following values:

```
id_number - 1712409137
name - "Liu Zhi Yu"
age - 19
college_name - "International Education"
std - 10
div - 'A'
```

```
S1.id_number = 1712409137;
strycpy(S1.name, "Liu Zhi Yu")
S1.age= 19
strycpy(S1.college_name, "International Education")
S1.std= 10
S1.div= 'A'
```

Question 3. Arrays and Pointers **[21 marks]**

(a) **[3 marks]** Three array declarations shown below have errors. Write the correct declaration statements.

```
# define N 5
int a[N] = {0, 2, 2, 3, 4, 5};
int b[N−6];
int c [3.0];
```

```
int a[N] = {0, 2, 2, 3, 4};
The initializer has 6 elements but the array has 5 spaces only (1
mark)
int b[N-6];
The array size is negative (ie N-6 = 5-6 = -1) (1 mark)
int c[3.0]
The array size is of data type float but it must be an integer
according
 to the int declaration (1 mark)
```

(b) **[3 marks]** What is the value of myArray[myArray[1]] based on the following array declaration?

```
int myArray[ ] = {0, 2, 4, 6, 8, 10};
```

```
Answer:  4
myArray[myArray[1]] -> myArray[2] -> 4
```

(Question 3 continued on next page)

(Question 3 continued)

(c) **[3 marks]** Give a declaration for the variable r in each of the following cases.

r is a pointer to a char.

```
char *r; (1 mark)
```

r is an array of 5 pointers to char.

```
char *r[5]; (1 mark)
```

r is a pointer to a function that takes no arguments and returns an int.

```
int (*r)(void); (1 mark)
```

(d) **[7 marks]** Given the following array and pointer declarations:

```
int ia[ ] = { 1, 2, 3, 4, 5, 7, 8, 9};
int *iap = &ia[0];
```

i. **[2 marks]** Write C language statements showing two different ways to access the value stored in the first element of the array ia

```
ia[0] (1 mark each)
*ia or *iap
```

ii. **[2 marks]** If the base address of the array ia is at (decimal) 1000, what is the value of iap+2? Assume that an int occupies 4 bytes.

```
1000 + 2*4 = 1000 + 8 = 1008
```

iii. **[3 marks]** Write a for-loop to iterate through the array outputting each element of ia using array indexes?

```
for (int i = 0; i < 9; i++) (2 marks)
printf("%d ", ia[i]); (1 mark)
```

(e) **[5 marks]** Complete the function below so that takes in an integer array and its length, returns 1 if the array stores **only** negative numbers, and returns 0 if the array stores at least one number that is not negative. (**Note**: 0 is **NOT** a negative number.)

```
int allNeg(int array[ ], int length) {








        for(int i=0; i< length; i++)
        {
            if (array[i]>0)){
                return 0;
        }
        return 1;





}
```

Question 4. Dynamic Memory Allocations **[10 marks]**

(a) **[6 marks]** What is the output of the following C program? Assume that the malloc is successful.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char *mem_allocation;
    mem_allocation = (char *) malloc( 20 * sizeof(char) );
    if( mem_allocation== NULL )
    {
        printf("Couldn't to allocate requested memory\n");
    }
    else
    {
        strcpy( mem_allocation,"https://www.wgtn.ac.nz/");
    }
    printf("Dynamically allocated memory content : %s\n", mem_allocation );
    free(mem_allocation);
}
```

```
Dynamically allocated memory content :  https://www.wgtn.ac
```

(b) **[4 marks]** What is the problem with following code?

```
#include<stdio.h>
#include <string.h>
int main()
{
    char *p = (char *) calloc(20, sizeof(char));
    if( p== NULL )
    {
        printf("Couldn't allocate requested memory\n");
       return 0;
    }
    strcpy(p,"Hello World!");
    while(*p != '\0')
        printf("%c",p++);
    free(p);
    return 0;
}
```

Memory Leak
The problem is memory leak, p is not pointing to the begining of the allocated block. free(p) cannot free the allocated memory

Question 5. Data Structures **[16 marks]**

(a) **[4 marks]** What is the output of the following C program?

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

struct employee
{
    int id;
    char name[50];
};

struct employee e1;

int main( )
{
   e1.id=101;
   strcpy(e1.name, "Wei Zi Yi");
    printf( "Employee 1 ID : %d\n", e1.id);
    printf( "Employee 1 Name : %s\n", e1.name);
   return 0;
}
```

```
Employee 1 ID : 101 (2 marks)
Employee 1 Name :  Wei Zi Yi (2 marks)
```

(b) **[2 marks]** In C, a node in a linked list is implemented using a structure. Declare a C structure with tag `node` that defines a node of a *doubly* linked list. For simplicity, declare the data field to be of type `int` with identifier `data`.

```
struct node{
 int data;
 struct node *next; // pointer to next element
 struct node *prev; // pointer to previous element};
};
```

(Question 5 continued on next page)

(Question 5 continued)

(c) **[10 marks]** Given the following struct declaration for a student, complete the Grades function which

- calculates each student's grade as 'P' or 'F' depending on whether the score is higher than or equal to 75, and
- returns the average class score;

The input parameter of Grades function is an array of students, each of which have an id and score, but do **not** yet have a grade.

```
#define SIZE 64

struct Student {
   unsigned int id;
   unsigned float score;
   unsigned char grade; // Must be 'P' or 'F'
};

typedef struct Student STyp;

float Grades(STyp class[SIZE]){

         int i;
         float sum=0;
         for(i=0; i<SIZE; i++){
            sum = sum + class[i].score;
            if(class[i].score >= 75){
               class[i].grade = 'P';
            } else{
               class[i].grade = 'F';
            }
         }
         return sum/SIZE;

}
```

Question 6. File Input/output **[7 marks]**

(a) **[3 marks]** Write a C statement that will open a text file `output.txt` for output, appending to existing contents (if any).

```
FILE *fp = fopen("output.txt", "a");
```

(b) **[4 marks]** Consider the following C code:

```
#include <stdio.h>

int main()
{
    char c;
    FILE *infp = fopen("infile.txt", "r");
    FILE *outfp = fopen("outfile.txt", "w");
    while( (c=getc(infp)) != EOF ) {
        putc(c-1, outfp);
    }
    fclose(infp);
    fclose(outfp);
    return 0;
}
```

If the contents of infile.txt is

```
gdkkn
```

What would be the contents of outfile.txt?

Hint: see the list of ASCII codes.

```
fcjjm
```

Question 7. Low-level Socket Programming [12 marks]

(a) **[2 marks]** The C source file hello.c contains the following:

```
#include <stdio.h>

int main(void)
{
   #ifdef Task1
      printf("Hello World!");
   #else
      printf("C programming!");
   #endif
   return 0;
}
```

If the source file is compiled with the command

```
gcc hello.c -o hello
```

What is the output when hello is run?

```
C programming!
```

(b) **[2 marks]** What are the two types of sockets supported by the socket system call?

```
Stream (TCP) and datagram (UDP)
```

(c) **[5 marks]** Briefly explain the steps of creating a socket in a **server** process. Please name the system call for each the step (if any).

```
Create a socket with the socket() system call.
Bind the socket to an address using the bind() system call.
Listen for connections with the listen() system call
Accept a connection with the accept() system call
Send and receive data
```

Question 8. Process Management **[10 marks]**

(a) **[2 marks]** Briefly explain the difference between a program and a process.

Program is static, with the potential for execution)
while Process is a program in execution and have a state
One program can be executed several times and thus has several processes

(b) **[8 marks]** You are given the following C program:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #include <unistd.h>
4   #include <sys/wait.h>
5
6   int gvar = 2;
7
8   int main(void)
9   {
10      int lvar = 4;
11      pid_t pid;
12
13      if ((pid = fork()) < 0) {
14          printf("fork error\n");
15      }
16      if (pid == 0) {
17          gvar++;
18          lvar++;
19      } else {
20          wait(NULL);
21          gvar++;
22      }
23
24      printf("%ld %d %d\n", (long)getpid(), gvar, lvar);
25      exit(0);
26  }
```

i. **[2 marks]** Which line(s) are run only in the parent process?

Lines 20 and 21

ii. **[6 marks]** Assume that the fork is successful. The parent process ID is 23476 and the child process ID is 23477. What is the output of the program?

```
23477 3 5
23476 3 4
```

* * * * * * * * * * * * * * *

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.

Reference Information

Some of this information might be useful while answering questions on the exam. Feel free to remove this page for reference while you work. **Please do not write on this page - anything written here will not be graded.**

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	