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

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

NWEN 241: Final Exam

2021, December 28 ** WITH SOLUTIONS **

Instructions

- Time allowed: **120 minutes**
- Attempt **all** the questions. There are **100 marks** in total.
- 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 unmarked paper Chinese-English translation dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	Marks	
1. C Fundamentals.	[14]	
2. Control Structures.	[24]	
3. Arrays, Characters and Strings.	[11]	
4. Arrays and Pointers.	[11]	
5. Derived Types and Dynamic Memory.	[20]	
6. File I/O.	[10]	
7. Process Management and Socket Programming	[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. Operators and Operator Precedence. **[14 marks]**

(a) **[2 marks]** Declare a macro symbolic constant `SPEED` with a single-precision floating point value 3.25×10^{-26}.

```
#define SPEED 3.25e-26f
```

(b) **[2 marks]** A C program contains the following declarations:

```
int i, j;
double ix;
short s;
float x;
```

What is the resulting **data type** of the expression?

```
x / s - 2.5 * 'Z' + ix * i / j
```

```
double
```

(c) **[4 marks]** What will be the output of the following program?

```
void main(){
    int i = 8, k = 24;
    k /= i;
    printf("%d \n", k);
    k += 10;
    printf("%d \n", k);
}
```

```
3
13
```

(Question 1 continued on next page)

(Question 1 continued)

(d) **[4 marks]** What will be the output of the following program?

```
#include <stdio.h>

int func( int a)
{
    return --a;
}

int main(void)
{
    int i = 5;
    int j = 6 * func( i );
    printf ("%d %d", i++, j);
    return 0;
}
```

5 24

(e) **[2 marks]** Re-write func(int a) in program in (d) into a function-like macro FUNC(A), such that when the call to func(i) in the program is replaced with FUNC(i), the outputs will remain the same.

```
#define Func(A) ((A)-1)
```

Question 2. Control Structures. **[24 marks]**

(a) **[4 marks]** Rewrite the following code using a for loop instead of while loop.

```
int doThings(int n) {
    int sum = 0;
    int count = 3;
    while (count <= n)
    {
        sum = sum + count;
        count ++;
    }
    return sum;
}
```

```
int doThings(int n) {
    int sum = 0;


    for (int count = 3; count <= n; count++)
    {
        sum = sum + count;
    }

    return sum;
}
```

(b) **[10 marks]** The function F(...) is defined below. What will be the output of calling F(10,3) and then F(10,2)?

```
int F(unsigned int m, unsigned int n) {

    int i = 0;
    while (i < m)
    {
        if (i % n)
        {
            printf("%d ", i);
        }
        i++;
    }
}
```

F(10, 3) → 1 2 4 5 7 8

F(10, 2) → 1 3 5 7 9

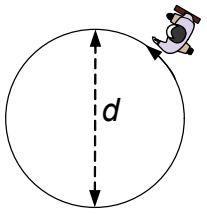
(Question 2 continued on next page)

(Question 2 continued)

(c) **[6 marks]** Complete this C program that calculates your average speed if you walk around a circle. The circle has a diameter.

- You must define a constant PI with the value of π (3.14) and you must use the constant in averageSpeed(...).
- main() must ask the user for the diameter of the circle and time of travel.
- averageSpeed(...) must calculate and return the average speed.

Hint: You need to calculate the average speed: $\frac{\pi \times Diameter}{time}$.




```
#include <stdio.h>

// The constant PI
#define PI 3.14


float averageSpeed(float d, float time){

    float averageSpeed = 0.0;
    averageSpeed = (PI * d)/time;
    return averageSpeed;



}

int main(){
    float diameter, time;

    printf("Enter diameter and time of travel: ");
    scanf("%f %f", &r, &time);


    printf("The average speed: %f", averageSpeed(diameter, time));

    return 0;
}
```

(Question 2 continued on next page)

(Question 2 continued)

(d) **[4 marks]** The function Calc(...) is defined below. What will be the output of calling Calc('+') and then Calc('=')?.

```
void Calc(char ch){
    int result = 12, a = 4;
    switch(ch)
    {
        case '-':
            result -= a;
            printf("Case A %d \n", result);
            break;
        case '+':
            result += a;
            printf("Case B %d \n", result);
            break;
        case '/':
            result /= a;
            printf("Case C %d \n", result);
            break;
        case '*':
            printf("Case D %d \n", result = a * a);
        default :
            printf("default %d \n", result);
    }
}
```

Calc('+') → Case B 16

Calc('=') → default 12

Question 3. Arrays, Characters and Strings. **[11 marks]**

(a) **[2 marks]** Using only one C statement, declare an array which can hold 50 integers with initial values 5, 6, 7 and 8 for the first four elements, and 0 for the remaining elements. Name this array iarray.

```
int iarray[50] = {1, 2, 3, 4};
```

(b) **[5 marks]** What will be the output of the following program?

Note: Assume a compiler where an integer is 4 bytes, a char is 1 byte and a pointer is 4 bytes.

```
int main() {
    int array1[ ] = { 1, 2, 3 };
    int* ptr1 = array1;

    char array2[ ] = "ABCDE\0WXYZ";
    char* ptr2 = array2;

    printf ("sizeof array1[] = %d ", sizeof(array1));
    printf ("sizeof ptr1 = %d ", sizeof(ptr1));

    printf ("sizeof array2[] = %d ", sizeof(array2));
    printf ("lenght of array2[]= %d ", strlen(array2));
    printf ("sizeof ptr2= %d ", sizeof(ptr2));

    return 0;
}
```

```
sizeof array1[] = 12
sizeof ptr1 = 4
sizeof array2[] = 11
length of array2[]= 5
sizeof ptr2= 4
```

(Question 3 continued on next page)

(Question 3 continued)

(c) **[2 marks]** An array has been declared as:

char myCharacters[ ]= {'a', 'b', 'c', 'd', 'r'};

Using sizeof(), write a C expression that calculates the number of elements of the array.

sizeof(myCharacters)/sizeof(char)

(d) **[2 marks]** Using a macro, declare a C string symbolic constant names STRING with value "Hello, world"

#define STRING "Hello, world"

Question 4. Arrays and Pointers. **[11 marks]**

Suppose x is an array of char, and we have just executed this code (a char is 1 byte):

```
char* sp;
for (i = 0; i < 6; i++)
    x[i] = 'a' + i;

sp = x;
char** spp = &sp;
```

Suppose that x[0] is stored at address 1950. What is the value of each of the following expressions? **[1 marks each]**

(**Hint:** see the list of ASCII codes)

x	1950
&x[0]	1950, same as x
*x	a, same as x[0]
x[2]	c
&x[2]	1952 (2 bytes from x)
sp+3	1953 (1 more byte)
*(sp+3)	d, same as x[3]
*sp+3	d, same as x[3]
*(x+3)	d, same as x[3]
*(&x[3] +2)	f
*spp + 3	1953

Question 5. Derived Types & Dynamic Memory. **[20 marks]**

(a) **[5 marks]** Consider the following C code:

```
enum maker { toyota, honda, mercedes = 10, audi };

union info {
    int year;
    char age;
};

struct car {
    enum maker maker;
    char model[10];
    union info info;
};
```

i. **[1 mark]** What is the value of the symbolic constant audi?

11

ii. **[2 marks]** Using C statements, declare a variable named c1 which is of type struct car, and initialize the members maker and model to honda and "x3".

struct car c1 = {honda, "x3"};

iii. **[2 marks]** Suppose that for the variable c1 declared in (ii), the following assignment statement is given:

```
c1.info.year = 2017;
```

What will be the value of c1.info.age? Briefly explain your answer.

c1.info.age will have invalid/garbage/rubbish value because

it shares storage with c1.info.year

(Question 5 continued on next page)

(Question 5 continued)

(b) **[4 marks]** Consider a singly-linked list which contains a list of integers. A node in this list is defined as follows:

```
struct node {
    int data;
    struct node *next;
};
```

Suppose that head points to the head of the list.

Write the necessary C code to print all the elements of the list, beginning at the head.

```
struct node *p = head;
while(p != NULL) {
    printf("%d", p->data);
    p = p->next;
}
```

(c) **[8 marks]** Consider the following struct,

```
typedef struct address_info {
    char name[50];
    char street [100];
    char city [50];
    char state [20];
    int post_code;
} Address;
```

Write a function with prototype

```
void printAddress(Address *addr);
```

that will print out the information of an address, i.e. name, street, city, state and post code.

```
void printAddress(Address *addr)
{

    printf( "name: %s\n", addr->name);
    printf( "street: %s\n", addr->street);
    printf( "city: %s\n", addr->city);
    printf( "state: %s\n", addr->state);
    printf( "post_code: %d\n", addr->post_code);




}
```

(Question 5 continued on next page)

(Question 5 continued)

(d) **[3 marks]** Given the following C program:

```
#include <stdio.h>

int func(int x)
{
    static int y;

    if (x == 0)
        y += 5;
    else if (x == 1)
        y += 10;
    else
        y++;

    return y;
}

int z;

int main(void)
{
    int w;

    func(1);
    func(2);
    w = func(z);
    printf("%d", w);

    return 0;
}
```

i. **[1 mark]** What is the initial value of y?

0

ii. **[1 mark]** In which memory segment is z stored?

Data segment

iii. **[1 mark]** What is the output of the program?

16

Question 6. File I/O. **[10 marks]**

Write a program in C that will read a file called myjourny.txt and print the contents to the standard output. It should print an error message if the file cannot be read.

```
int main(){

        FILE *fptr;
        char str;

        fptr = fopen("myjourny.txt", "r");
        if (fptr == NULL)
        {
            printf(" File does not exist or cannot be opened.\n");
            exit(0);
        }
        str = fgetc(fptr);
        while (str != EOF)
        {
            printf ("%c", str);
            str = fgetc(fptr);
        }
        fclose(fptr);









    return 0;
}
```

Question 7. Process Management and Socket Programming. **[10 marks]**

(a) **[1 mark]** Name the family of system calls that does not return control back to the calling point on termination.

exec() system call

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

SOCK_DGRAM and SOCK_STREAM

(c) **[2 marks]** How many times will the following C progrm print Hi?

```
#include<stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    fork ();
    fork ();
    fork ();
    printf ("Hi\n");
}
```

8

(Question 7 continued on next page)

(Question 7 continued)

(d) **[5 marks]** You are given the following C program:

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

int gvar = 20;

int main(void)
{
    int lvar = 40;
    pid_t pid;

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

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

Answer the following TWO (2) questions:

i. **[3 marks]** What does the wait(NULL) function do?

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

ii. **[2 marks]** Assume that the `fork()` is successful and that the parent process ID is 16232 while the child process ID is 16233. What is the output of the program?

16233 21 41

16232 20 40

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

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	