Family Name: ....

Other Names: ....

Signature: ....

Student Number: ....

THE UNIVERSITY OF NEW SOUTH WALES
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Sample Examination

COMP 1917 Computing 1

EXAM DURATION: 2 HOURS
TOTAL MARKS FOR THIS PAPER: 60
THIS PAPER COUNTS FOR 40% OF YOUR FINAL GRADE.
THIS PAPER HAS 14 PAGES INCLUDING THIS COVER PAGE.
THIS PAPER MUST NOT BE RETAINED BY CANDIDATE.
APPROVED MATERIALS: UNSW APPROVED CALCULATOR.
ALL QUESTIONS MAY BE ATTEMPTED.

Question 1. [3 marks]

Consider this C program:

```
#include <stdio.h>

int main( void )
{
    int x=1;
    int y=0;

    while( x < 100 ) {
        x = x + 8 * y++ ;
        printf("%d\n", x );
    }

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 2. [4 marks]

Consider this C program:

```
#include <stdio.h>

int main( void )
{
    int *p, *q;
    int  x, y;

    x = 7;
    y = 8;

    q = &x;
   *q = 10;
    p = q;

    y = *p + *q;
   *p =  x + y;

    printf("x = %d, y = %d\n", x, y );

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 3. [4 marks]

Consider this C program:

```
#include <stdio.h>


int df( int n )
{
    if( n < 2 ) {
        return( 1 );
    }
    else {
        return( n * df( n-2 ));
    }
}

int main( void )
{
    printf( "5!! = %d\n", df( 5 ));
    printf( "6!! = %d\n", df( 6 ));

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 4. [3 marks]

The following function is supposed to take a positive integer `n` and an array `a[]` of `n` integers, and sort the `n` items in the array into increasing order. However, the function as currently written does not accomplish the sorting correctly, because there are three programming errors in the code. For each error, identify the number of the line on which it occurs, and indicate how you would correct it.

```
 0 void selectionSort( int n, int a[] )
 1 {
 2    int i, j, min, tmp;
 3
 4    for( i=0; i < n; i++ ) {
 5       min = i; // initial minimum is first unsorted item
 6
 7       // find index of minimum item
 8       for( j = i+1; j < n; j++ );
 9           if( a[j] < a[min] )
10               j = min;
11
12       // swap minimum item into place
13       tmp = a[i];
14       a[min] = tmp;
15       a[i] = a[min];
16    }
17 }
```

Question 5. [4 marks]

Consider this C program:

```
#include <stdio.h>

void print_edge( int k )
{
    int i;

    for( i=0; i < k; i++ ) {
        printf( "+-" );
    }
    printf( "+\n" );
}

void print_legs( int k )
{
    int i;

    for( i=0; i < k; i++ ) {
        printf( "| " );
    }
    printf( "|\n" );
}

int main( void )
{
    int k;

    for( k=0; k < 4; k++ ) {
        print_edge( k );
        print_legs( k );
    }
    print_edge( k );

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 6. [3 marks]

Consider this C program:

```
#include <stdio.h>

void f( char *s )
{
    if( !*s ) {
        return;
    }
    f( s+1 );
    putchar( *s );
}

int main(void)
{
    f("kernighan");
    putchar('\n');

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 7. [4 marks] Consider this C program:

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

struct node {
    int value;
    struct node *next;
};

struct node *new( int val )
{
    struct node *n = malloc(sizeof(struct node));
    n->value = val;
    n->next  = NULL;
    return n;
}

struct node *insert( struct node *n, struct node *h )
{
    n->next = h;
    return n;
}

int main( void )
{
    struct node *a, *b, *c;
    int i;

    a = new(1);
    b = insert( new(5), insert( new(2), a ));
    c = insert( new(3), b->next );
    a = insert( a, c );

    for( i=0; i < 5; i++ ) {
        printf(" %d", b->value );
        b = b->next;
    }
    printf("\n");

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

Question 8.

a) [2 marks] Draw the Binary Search Tree that would result from inserting these numbers, in order, into an initailly empty tree:

$$5, 3, 8, 7, 1, 4, 6, 2, 9$$

b) [3 marks] State breifly (one or two lines) what the following function does.

```
char *mystery( char *a, char *b )
{
    char *c = a;

    while ( *c != ’\0’ ) {
        c++;
    }
    while (( *c++ = *b++ ) != ’\0’ )
        ;

    return a;
}
```

What is the name of the C standard library function that fulfils the same purpose as `mystery()` ?

Question 9. [4 marks] Consider this C program:

```
#include <stdio.h>

int a = 1;
static int b = 1;

int f( int c )
{
    static int d = 1;
    int e = 0;

    a++;
    b += d;
    c = c + 2;
    d = d + a - b + c;
    e = e + 2*d + 1;
    return( e+2 );
}

int main( void )
{
    int a, d;
    a = 3;

    for( d=0; d < 3; d++ ) {
        printf("%d\n", f(a));
    }
    printf("%d\n", a );
    printf("%d\n", b );
    printf("%d\n", d );

    return 0;
}
```

The program is valid C. It executes without error.
Indicate clearly and exactly what output will be printed.

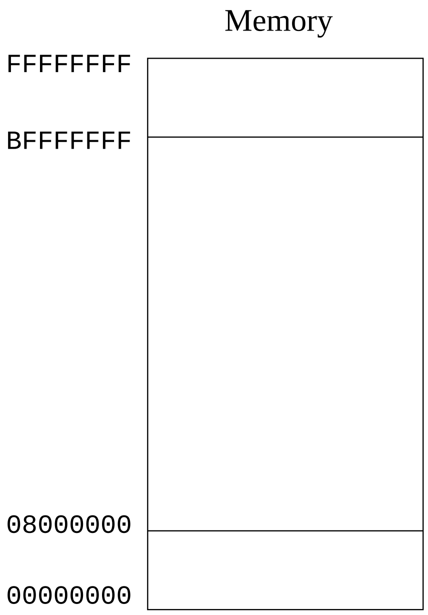
Question 10. [4 marks]

Copy the diagram below into your script book and use it to draw a Memory Map showing where various kinds of variables declared in a C program would be stored in a computer's memory.

Given the following variable declarations, indicate which region of memory the compiler would use to store the variables `k`, `l` and items in array `a[]`, when the program executes.

```
{
  static int k;
  int  l;
  int *a =(int *)malloc( 10 * sizeof( int ));

  ...
}
```



Question 11. [8 marks]

A 2-dimensional array is called a **magic square** if the sum of the numbers along any row, column or diagonal is the same.
(For example, the following is a magic square because the numbers along any of the 4 rows, 4 columns or 2 diagonals is 34).

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

Write a function `int is_magic(int a[4][4])` which takes as parameter a 4×4 array of integers and returns `1` if the numbers in the array form a magic square, `0` otherwise.

Question 12. [6 marks]

a) Convert the following binary number to decimal.

10101.001

b) Convert the following decimal number to binary.

45.375

c) Assume that signed binary numbers are stored in two's complement form, in 8 bits (for example, $11111111_2 = -1_{10}$).

(i) Convert the following binary number to decimal.

01101011

(ii) What is the negative of the number from part **c(i)**, written in two's complement (binary) form?

(iii) Use your answer from part **c(ii)** to compute the result of the following binary subtraction. Write your answer in two's complement form.

$00110001 - 01101011$

Question 13. [8 marks]

The instruction set for the simple machine presented in lectures is given in the table on page 14. The following three sub-questions refer to the program shown above the table (also on page 14). All addresses and memory contents are given in Hexadecimal (base 16) notation.

a) What would be in registers R1, R2, R3 and R4 after running the program starting at address `A0`, given that the relevant memory contents are as shown?

b) Suppose that the contents of memory location `81` is changed from `02` to `03`. What would now be in register R3 after running the above program starting at address `A0`?

c) Write a segment of C code that accomplishes the same task as the machine language program given on page 14.

Machine Language Program

```
 Address  Contents

    80       00
    81       02

     .      . .

    A0      20 00
    A2      21 01
    A4      12 81
    A6      23 01
    A8      B2 B4
    AA      50 01
    AC      40 34
    AE      53 34
    B0      53 34
    B2      B0 A8
    B4      33 80
    B6      C0 00
```

Opcode	Operand	Description
1	RXY	LOAD register R with bit pattern found in memory cell whose address is XY
2	RXY	LOAD register R with the bit pattern XY
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY
4	0RS	COPY the bit pattern found in register R to register S
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations, and leave the result in register R
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating point notation, and leave the floating point result in register R
7	RST	OR the bit patterns in registers S and T and place the result in register R
8	RST	AND the bit patterns in registers S and T and place the result in register R
9	RST	EXCLUSIVE-OR the bit patterns in registers S and T and place the result in register R
A	R0X	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low order end to the high order end.
B	RXY	JUMP to the instruction located in the memory cell at the address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution.
C	000	HALT execution

[END OF EXAM]