SE185: Introduction to Computer Engineering and Problem Solving I
Midterm 1: Thursday (10-07-21)

Last Name: Tack First Name: Artemis Lab Section: 2

1. True/False Questions (10 x 1p each = 10p)

(a) The following statement is valid: char Exam1[2][2] = {{'2', '4'}, {'A', 'B'}};	[✓ TRUE]	/	FALSE
(b) The following statements are valid: int a [3]; int b [4]; a[1] = b[2];	[✓ TRUE]	/	FALSE
(c) A **long int** type variable requires eight bytes of memory	TRUE	/	[✓ FALSE]
(d) The **rand()** function is defined in **math.h**	TRUE	/	[✓ FALSE]
(e) A string is a char array but a char array is not a string.	~~TRUE~~	/	[✓ FALSE]
(f) This is a valid variable name in C: **_while**	TRUE	/	[✓ FALSE]
(g) This for loop iterate 5 times: for (int i=0; i<5; i--) { printf ("Hi"); }	[✓ TRUE]	/	FALSE
(h) If x=5 and y=7, the following while loop is an infinite loop: while (x != y){ printf("Hello!\n");}	[✓ TRUE]	/	FALSE
(i) The following statements are valid: char a [3]; char b [3]; a = b;	TRUE	/	[✓ FALSE]
(j) The following statement will give a compilation error: char alphabet [12] = "_SE_185";	TRUE	/	[✓ FALSE]

2. Expressions and Assignments (5 x 3p each = 15p)

What is the output of **each** of the **printf** statements below (there are **5 printf**)?

```
#include<stdio.h>
int main(){
        float result;
        int num1 = 4, num2 = 16, num3 = 12;
        double a = 3.0, b = 2, c = 4.0;
        printf ("(a) %f\n", result = num2/num1);
        printf ("(b) %0.2f\n", result = b+(num2%num1));
        printf ("(c) %0.1f\n", result = (num2*num3)/c);
        printf ("(d) %0.2f\n", result = num2/(5%num1));
        printf ("(e) %0.2f\n", result = ++num2+a*num3);
        return 0;
}
```

16 4 16/4 = 4

16 4 → 0 2 + 0 2.00

16 · 12 → 192/4.0 = 48

5 % 4 → 1 16/1 = 16

1 + 16 + 3.0 · 12

~~17 + 3 → 20 · 12~~ 1 + 16 + 36 3.0 · 12 17 + 36 = 53

Output of printf # 1: 4

Output of printf # 2: 2.00

Output of printf # 3: 48.0

Output of printf # 4: 16

Output of printf # 5: 53

3. Number Conversions (4 x 5p each = 20p)

(a) Convert 173_{10} to Binary

~~128+64 = 14~~

128+32

160+8 168+4 = 172 = 173

~~173/2 =~~

128 64 32 16 8421

0 000 0000 0000

1010 1101

1010 1101

(b) Convert $7A2F_{16}$ to Octal

~~0111 1010 0010 1111~~

~~3 6~~

0010 1111

0111 1010 0010 1111   0111 1010 0010 1111

~~011 110 100 010~~ 111

111

0 111 101 000 101 111   0 111 101 000 101 111

0 7 5 0 5 7   0 7 5 0 5 7

$\boxed{17257}$

(c) Convert 1001011_2 to Hexadecimal

1001 011

0100 1011

4 B

4B

(d) Convert 1100 1001_2 to Decimal

128 64 32 16 8 4 2 1

1 1 0 0 1 0 0 1

128+64 +8+1

192 9

201

4. Code Snippets (20p)

(a) **Code Snippets (3-6 lines maximum) (6p)** Write a while loop to the print odd numbers from 12 to 89. The output should be in the following format: 13, 15, 17, 19, ...

```
int x =12;
int num = 13;
   while (x >= 12 && x <= 89) {
if(num%2==1){printf("%d, ", num); }
       ++x;
       ++num;
   }
```

(b) **Code Snippets (3-12 lines maximum) (8p)** Write a short code that ask the users to enter 4 integer numbers, and then print the even numbers entered by the users.
If the user inputs 12 13 14, 15, the output should be in the following format: 12 14

```
int  num[4];
printf("enter 4 intigers");
 scanf("%d %d %d %d", num[0], num[1], num[2], num[3]);
  for(int i=0; i<4; i++){
       if(num[i] % 2 == 0){
            printf(num[i]);
       }
  }
```

(c) **Code Snippets (3-6 lines maximum) (6p)** Write a while loop that continuously ask the user to enter a floating-point number and print the number. [declare any variable if necessary].

```
  float float;
 while(1) {
      printf("enter a float num");
      scanf("%f", float);
      printf("%f", float);

 }
```

5. Rewriting Code (5 x 2p each = 10p)

(a) What is the output of the **printf** statements below.

```
#include<stdio.h>
#include<string.h>
void main(){
                          0                                   2 9
        char str1[50]="I am ready for SE 185 midterm 1";
        int i, j, count=0;
        while (i< strlen(str1)) {
              if(str1[i] == 'a' || str1[i] == 'e' || str1[i] == 'i' || str1[i] == 'o' || str1[i] == 'u') {
                  count++;
              }                      i++, i
              i++;
        }
        printf("\nCount = %d\n" , count);
}
```

6 is the output

(b) Rewrite the above C program by using for loop.

```
for (i=0; i< strlen( str1); ++i) {
   if (. . . . . . . . .) {
      count ++;
   }
}
```

6. Find program output (5p)

What is the output of the **printf** statements below?

j

(0,0)
10 12 14
i 11 13 15
16 21 55 (3,3)

r c

```
#include <stdio.h>

void main () {
        int userNumber [3][3] = {{10, 12, 14}, {11, 13, 15}, {16, 21, 55}};
        int i, j;
        for(i=0; i<3; ++i) {
                for(j=0; j<3; ++j){
                        if ((userNumber[i][j])%3==0){
                                printf("%d ", userNumber[i][j]+1);
                        }
                }

        }
}
```

if num in row & col is %3 then num +1

12 -> 13
15 -> 16
21 -> 22

13 16 22 is the output

7. Find program output (10p)

What is the output of the **printf** statements below?

```
#include<stdio.h>
#include<string.h>
void main(){
        char alpha [13] = "SE_185";
        char beta [13];
        int i;
        for(i=0; i<strlen(alpha); i++){
                if (alpha[i]=='_'){
                        alpha[i]='@';
                        continue;
                }
                printf("%c", alpha[i]);
        }
        printf("\n%s", alpha);
        printf("\n%s", strncpy(beta, "Rocks_ISU", 5));
                                      012345
        printf("\n%s", strcat(alpha, beta));
        printf("\n%.3s", alpha);
}
```

Output of printf # 1: S E @ 185

Output of printf # 2: S E @ 185

Output of printf # 3: ROCKS_

Output of printf # 4: S E @ 185 ROCKS_

Output of printf # 5: S E @X

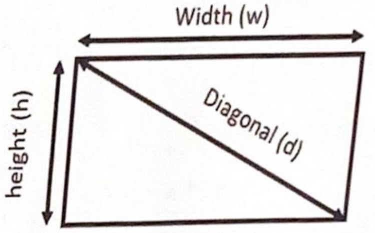
8. Debugging (10p)

Find and fix all the bugs/errors in the following C program and rewrite the code. Write a comment for each line that has bug and how to fix it.

```
#include<stdio.h>
               b
#include<stdlip.h>    // <stdlib> not <stdlip>

                         -> a is undeclared, should be int a
        double
        doable calculate(a);   // double not doable

        {

                int result = 42+a;

                return result;

        };

                                       using gcc
int
void main(){     //  at least on mac...  must be int not void
                         ->//probable meant \n not /n
            f
        print("Hello World!/n'); // double not single quotations

        double res = calculate(7);
                        lf
        printf("res=%d",res);   //  %lf for double type variables
                      (needs a comma

}
```

9. Bonus question: Rectangle Properties (15p)

Write a complete C program that asks the user to input the height and width of a rectangle and then call 3 individual functions to calculate: (1) area of the rectangle; (2) perimeter of the rectangle; and (3) diagonals of the rectangle. The main function must then print these values.



Area = w*h

Perimeter = 2(w + h)

$$Diagonal = \sqrt{(w^2 + h^2)}$$

```
# include <math.h>   # include <stdio.10.h>
area(int w, int h); {
    double ~~ret~~ areaAns = w·h;
    return areaAns;

}
Perimiter (int w, int h) {
    double perimiterAns = 2·(w+h);
    return perimiterAns;

}
diagonal (int w, int h) {
    double diagAns = sqrt(pow(w,2) + pow(h,2));
    return diagAns;

}


int main() {

  int w
  int h

  printf ("enter the height & width of a triangle: ")"

  scanf("%d %d", h, w);


 printf ("%d", area(w, h);

 printf ("%d", perimiter (w, h);

 printf ("%d", diagonal (w, h);

}
```