**CS 3335**

**HW5**

**25 points**

1. True or False: (2pts)

1. In a recursion function the base case could be empty. [TRUE/FALSE]

**Answer: FALSE**

1. Usually recursion is less efficient in terms of both execution speed and memory usage compared to iteration.[TRUE/FALSE]

**Answer: TRUE**

1. What does the following recursive function do? You have to identify what it computes. What will be the value of result if b=7? (2 + 1 = 3pts)

int find(int b)

{

if (b==0) /\*Base case \*/

return 0;

else /\*General case \*/

return (b + find(b - 1));

}

**Answer: It computes the sum of the first n natural numbers**

**Output: 7+6+5+4+3+2+1=28**

1. What will be the output of the following program: (2pts)

#include<stdio.h>

int main()

{

int array[]={45, 67, 89, 44, 88, 34, 65};

int \*array\_ptr = &array[3];

printf("%d\n", array\_ptr[3]);

return 0;

}

**Output: 65**

4. What will be the output of the following program: (3pts)

#include <stdio.h>

int main ()

{

int foo = 2;

int bar = 3;

int \* ptr;

ptr = &bar;

\*ptr = 5;

printf ("foo = %d\n", foo);

printf ("bar = %d\n", bar);

ptr = &foo;

printf ("foo = %d\n", foo); printf ("bar = %d\n", bar);

printf ("Value at ptr = %d\n", \*ptr);

return 0;

}

**Output:**

**foo = 2**

**bar = 5**

**foo = 2**

**bar = 5**

**Value at ptr = 2**

5. Write the output of the following program. (4 pts)

islower() checks if character is lowercase letter

toupper() converts lowercase letter to uppercase

#include <ctype.h>

#include <stdio.h>

void Encrypt(char T[])

{

for (int i = 0; T[i] != '\0'; i++)

{

if (T[i] == 'a' || T[i] == 'e')

T[i] = '#';

else if (islower(T[i]))

T[i] = toupper(T[i]);

else if (isdigit(T[i]))

T[i] = '@';

else

T[i] = '\*';

}

}

int main()

{

char text[]="C C0DIng Is Great8";

Encrypt(text);

printf("%s\n", text);

return 0;

}

**Output: \*\*\*@\*\*NG\*\*S\*\*R##T@**

6. Write the output of the following program if the input is – We are doing C programming. (2pts)

#include<stdio.h>

int main( )

{

char str[80];

printf( "Enter a string: ");

gets(str);

int words = 0;

for(int i = 0; str[i] != '\0'; i++)

{

if (str[i] == ' ')

{

words++;

}

}

printf("%d\n", words);

return 0;

}

**Output: We are doing C programming**

**4**

7. State true or false (4pts)

1. strcmp("ten", "tent") is greater than 0.

**Answer: FASLE -1**

1. strcpy(a,b) copies a to b, where a and b are two strings.

**Answer: FALSE copies b to a**

1. strlen(“I’m fine”) is 7.

**Answer: FALSE 8**

1. If we create a character array, str1, of length 5, strcpy(str1, “abcdef”) can result in buffer overflow.

**Answer: TRUE the limit is size 5**

8.

struct Point

{

int x, y;

};

struct Line

{

struct Point a, b;

};

int main()

{

struct Line plane[10];

}

State True/False for the above declarations (4pts)

1. plane is a struct

**Answer: TRUE creates an array of structures**

1. plane[6] is an array

**Answer: FALSE points to an element structure in the array**

1. plane[6].a is a struct

**Answer: FALSE it refers to a member of the structure**

1. plane[6].a.x is a struct

**Answer: FALSE it refers to a member of the structure**

9. Print the y coordinate of the point on the 6th line for the example in problem 8. (1pt)

**Answer: printf(“%d”, plane[6].a.y);**