WEEK-END ASSIGNMENT-08
C Storage Classes & Recursion
Operating Systems Workshop (CSE 3541)

Problem Statement:

Working with different storage classes and Experiment with one of the powerful tool, recursion, in problem solving and programming.

Assignment Objectives:

To learn about storage classes and get the idea of how function calls itself to solve computational problem.

Instruction to Students (If any):

Students are required to write his/her own program by avoiding any kind of copy from any sources. Additionally, They must be able to realise the outcome of that question in relevant to systems programming. You may use additional pages on requirement.

Programming/ Output Based Questions:

1. Consider the following ANSI C program;

```
#include <stdio.h>
int main()
{
        static int i=5;
        if(--i){
                main();
                printf("%d ",i);
        }
        return 0;
}
```

What is the output of the above program?

Output with explanation

```
0 0 0 0
```

2. Consider the following ANSI C program;

```
#include <stdio.h>
int a, b, c = 0;
void prtFun(void);
int main()
{ static int a = 1; /* Line 1 */
        prtFun( );
        a+=1;
        prtFun( );
        printf("\n %d %d ", a, b);
        return(0);
}
void prtFun(void)
{ static int a = 2; /* Line 2 */
        int b = 1;
        a + = ++b;
        printf(" \n %d %d ", a, b);
}
```

What is the output of the above program?

Output with explanation

```
4 2
6 2
2 0
```

3. Consider the following ANSI C program;

```
#include <stdio.h>
int a, b, c = 0;
void prtFun(void);
int main()
{ auto int a = 1; /* Line 1 */
        prtFun( );
        a+=1;
        prtFun( );
        printf("\n %d %d ", a, b);
        return(0);
}
void prtFun(void)
{ register int a = 2; /* Line 2 */
        int b = 1;
        a + = ++b;
        printf(" \n %d %d ", a, b);
}
```

What is the output of the above program?

Output with explanation

```
4  2
4  2
2  0
```

4. What is printed by the following ANSI C program? **[GATE 2005]**

```
#include<stdio.h>
int f(int n, int k){
        if(n==0) return 0;
        else if(n%2) return f(n/2, 2*k)+k;
        else return f(n/2, 2*k)-k;
}
int main(){
        printf( "%d",f(20,1));
        return 0;
}
```

Output with explanation

```
9
```

5. What is printed by the following ANSI C program? **[GATE 2007]**

```
#include<stdio.h>
void f(int n){
        if(n<=1){
                printf("%d",n);
        }
        else{
                f(n/2);
                printf("%d",n%2);
        }
}
int main()
{
        f(173);
        return 0;
}
```

Output with explanation▼

```
10101101
```

6. What is printed by the following ANSI C program? **[GATE 2011]**

```
#include<stdio.h>
unsigned int foo(unsigned int n, unsigned int r)
{
        if(n>0) return((n%r)+foo(n/r,r));
        else return 0;
}
int main()
{
        printf("%d\n",foo(345,10));
        return 0;
}
```

Output▼

12

7. What is printed by the following ANSI C program?

```
#include<stdio.h>
void print(int n)
{
        printf("Hello ");
        if(n++ == 0) return ;
        print(n);
        n++;
}

int main()
{
        print(-4);
}
```

Output▼

Hello Hello Hello Hello Hello

8. Consider the program below;

```
#include<stdio.h>
int foo(int a){
        if( a == 0 || a ==2) return 1;
        return (foo(--a) * (a--));
}
int main()
{
        printf("%d\n",foo(4));
}
```

Output▼

6

9. Consider the following C program

```
#include<stdio.h>
int main(){
        register int a =10;
        int *ptr = NULL;
        ptr = &a;
        *ptr = 5;
        printf("%d",*ptr);
        return(0);
}
```

Find the error in the program with proper reasoning

Output▼

The corrected code snippet declares "a"
as a regular "int" variable instead of a "register"
variable, allowing it to be assigned a memory address.

10. Consider the following C function;

```
file1.c
---------------
extern in count;
void write_extern(){
        count +=2;
}
```

```
file2.c
---------------
#include<stdio.h>
#include "file1.c"
int count = 5;
int main(){
        write_extern();
        write_extern();
        printf("%d\n", count);
        return(0);
}
```

Find the output if "**`file2.c`**" is compiled and executed:

Output with explanation▼

11. Write the output of the following program;

```
#include<stdio.h>
int i=5;
int main()
{
        extern int j;
        printf("\ni=%d \nj=%d",i,j);
        int j=10;
        return 0;
}

int j =10;
```

Output▼

error in this code because j is declare 2 times

if we remove j declaration once
i.e. remove "int"
form (int j=10;)

then o/p :- i=5
j=10

12. Find the output and different types of pointer involved in the code snippet;

```
#include <stdio.h>
int fun(int num){
        while(num>0)
                num=num*fun(num-1);
        return num;
}
int main(){
        int x=fun(8);
        printf("%d",x);
        return 0;
}
```

Output▼

0 (there is no base case)

There are no explicit pointers

13. Write a program to find the sum of an array elements using recursion.

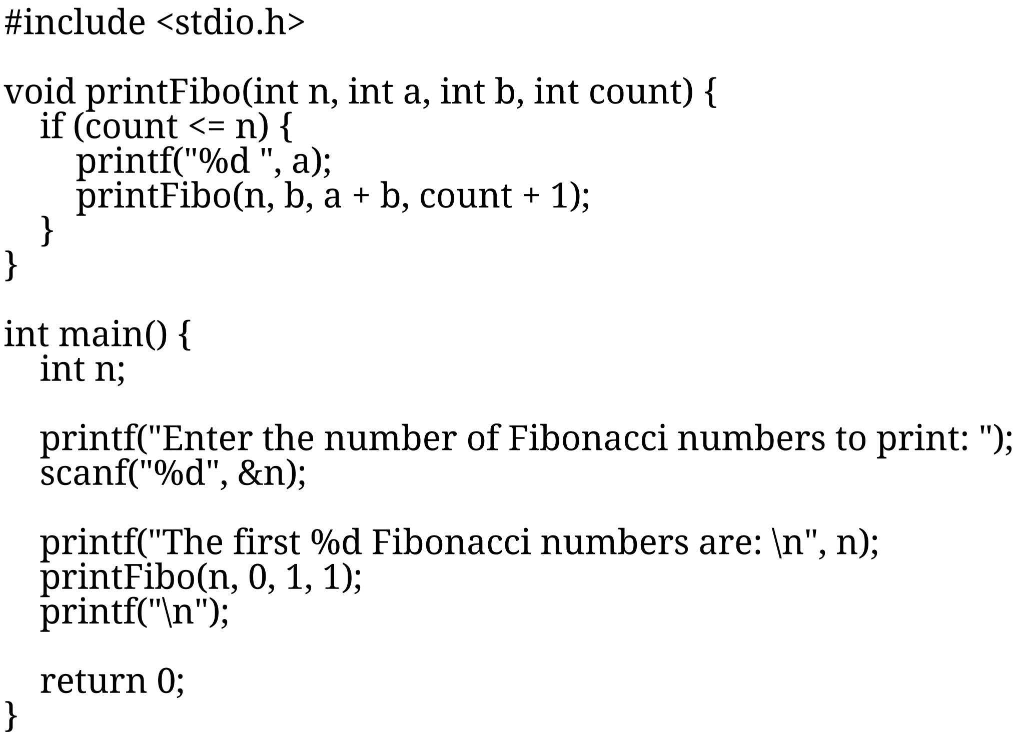
Program and Output▼

```
#include <stdio.h>

int arraySum(int arr[], int size) {
    if (size <= 0) {
        return 0;
    } else {
        return arr[size - 1] + arraySum(arr, size - 1);
    }
}

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    if (size <= 0) {
        printf("Please enter a valid array size.\n");
        return 1;
    }

    int arr[size];

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    int sum = arraySum(arr, size);

    printf("The sum of the array elements is: %d\n", sum);

    return 0;
}
```

14. Write a program to print “**n**” Fibonacci numbers using recursion.[N.B: The program format should be as follows]

```
#include <stdio.h>
... print_fibo(......){
        ...
        ...
        ...
}
... main(){
        // get data from user
        print_fibo(...); // to print elements
}
```

Program and Output▼



```
#include <stdio.h>

void printFibo(int n, int a, int b, int count) {
   if (count <= n) {
      printf("%d ", a);
      printFibo(n, b, a + b, count + 1);
   }
}

int main() {
   int n;

   printf("Enter the number of Fibonacci numbers to print: ");
   scanf("%d", &n);

   printf("The first %d Fibonacci numbers are: \n", n);
   printFibo(n, 0, 1, 1);
   printf("\n");

   return 0;
}
```

15. Write a program to print the binary equivalent of a Decimal number using recursion.

Program and Output▼

```
#include <stdio.h>

void decimalToBinary(int decimal) {
    if (decimal > 0) {
        decimalToBinary(decimal / 2);
        printf("%d", decimal % 2);
    }
}

int main() {
    int decimalNumber;

    printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);

    if (decimalNumber < 0) {
        printf("Please enter a non-negative decimal number.\n");
        return 1;
    }

    printf("Binary equivalent of %d is: ", decimalNumber);

    if (decimalNumber == 0) {
        printf("0");
    } else {
        decimalToBinary(decimalNumber);
    }

    printf("\n");

    return 0;
}
```

16. Write a program to remove adjacent duplicate characters from a string using recursion.

Program and Output▼

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

void removeAdjacentDuplicates(char *str) {
    if (str[0] == '\0' || str[1] == '\0') {
        return;
    }
    if (str[0] == str[1]) {
        int i = 0;

        while (str[i] != '\0') {
            str[i] = str[i + 1];
            i++;
        }
        removeAdjacentDuplicates(str);
    } else {
        removeAdjacentDuplicates(str + 1);
    }
}

int main() {
    char inputString[100];

    printf("Enter a string: ");
    scanf("%s", inputString);

    printf("String before removing adjacent duplicates: %s\n", inputString);

    removeAdjacentDuplicates(inputString);

    printf("String after removing adjacent duplicates: %s\n", inputString);

    return 0;
}
```

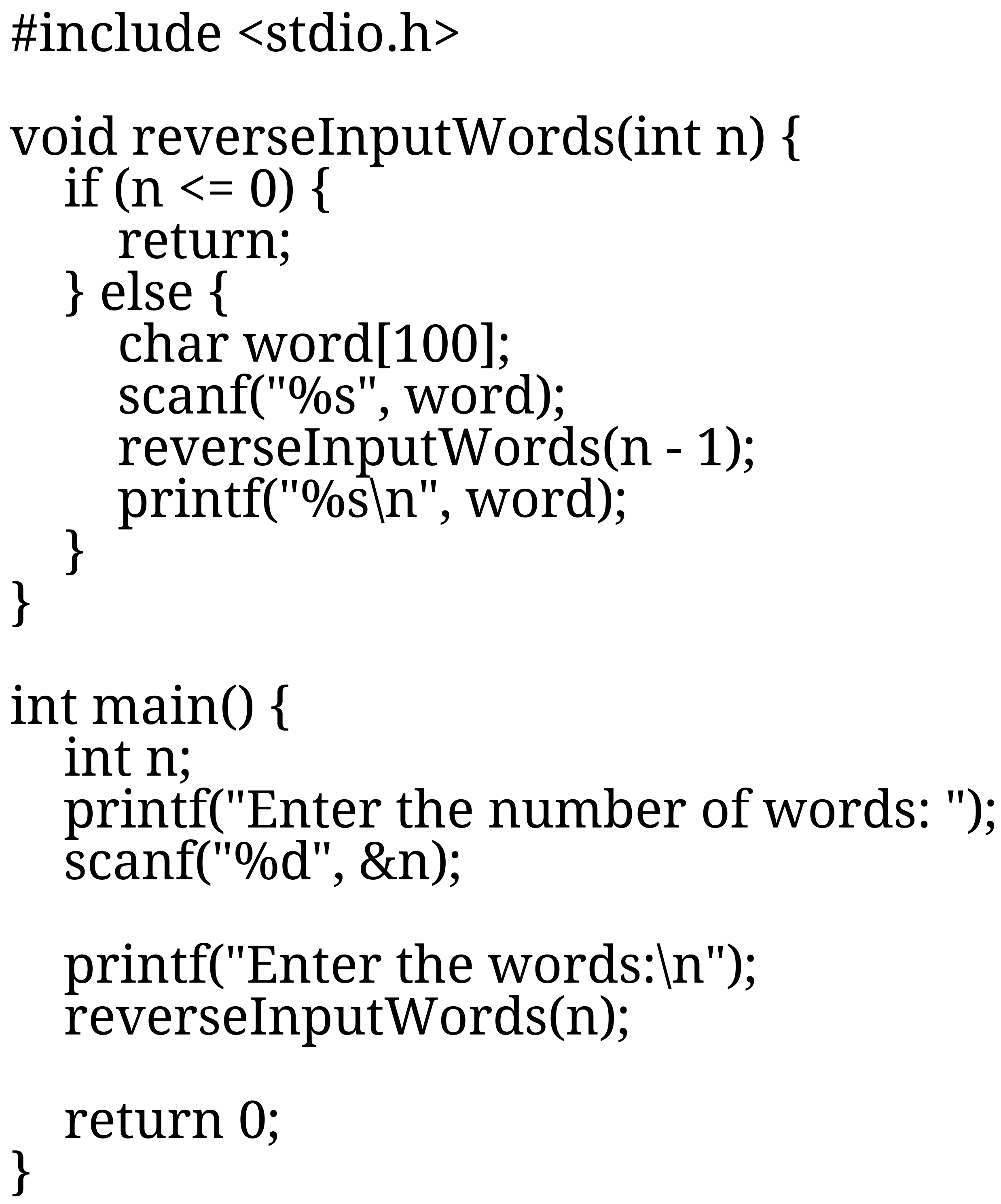
17. Write a program to find the sum of a geometric sequence using recursion.

Program and Output▼

```
#include <stdio.h>

double geometricSum(int firstTerm, double ratio, int terms) {

   if (terms <= 0) {
      return 0;
   } else {

      return (firstTerm + geometricSum(firstTerm * ratio, ratio, terms - 1));
   }
}

int main() {
   int firstTerm, terms;
   double ratio, sum;

   printf("Enter the first term of the sequence: ");
   scanf("%d", &firstTerm);

   printf("Enter the common ratio of the sequence: ");
   scanf("%lf", &ratio);

   printf("Enter the number of terms in the sequence: ");
   scanf("%d", &terms);

   sum = geometricSum(firstTerm, ratio, terms);

   printf("The sum of the geometric sequence is: %.2lf\n", sum);

   return 0;
}
```

18. Write a recursive function that takes n words as input and print them in reverse order on separate lines. The prototype of the function should be as follows:
    `void reverse_input_words(int n);`

Program and Output▼



```
#include <stdio.h>

void reverseInputWords(int n) {
    if (n <= 0) {
        return;
    } else {
        char word[100];
        scanf("%s", word);
        reverseInputWords(n - 1);
        printf("%s\n", word);
    }
}

int main() {
    int n;
    printf("Enter the number of words: ");
    scanf("%d", &n);

    printf("Enter the words:\n");
    reverseInputWords(n);

    return 0;
}
```