**Data Structure**

**Tutorial 2:**

**Memory Allocation**

**1. For a given C code and mentioned scenarios, what will be the output with respect to the memory layout in C? Explain with reason. (**Note: Use **size** command)

A.) Original File

Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*// Default Code Given by Mam*

*#include* <stdio.h>

int sum(int n);

int main()

{

    int number, result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

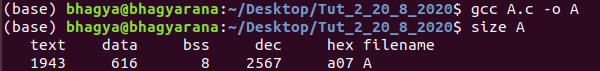
*return* n + sum(n - 1);

*else*

*return* n;

}

Original Code Size:



1.) When variable “number” is declared as a Global variable

Q1 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

*// When variable “number” is declared as a Global variable*

int sum(int n);

int number = 10; *// Intialised Global Variable Stored in data segment*

*//int number;  // Unintialised Global Variable Stored in bss segment*

int main()

{

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

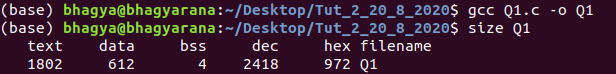
*return* n + sum(n - 1);

*else*

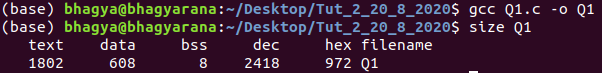
*return* n;

}

Q1 Initialized Code Size:



Q1 Uninitialized Code Size (Uncommenting Second Line):



Reason: We can clearly observe a difference of 4 bytes due to the fact that:

*Initialized Global Variable* is stored in **data segment** and *Uninitialized Global Variable* is stored in **BSS Segment**. Therefore, *Global Variables are stored data Segment or BSS depending on whether they are initialized or not*.

2.) When variable “number” is declared as a Static variable

Q2 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

*//When variable “number” is declared as a Static variable*

int sum(int n);

int main()

{

    static int number = 5; *// Itialised Static Variable Stored in data segment*

*//static int number; //   Unitialised Static Variable Stored in bss segment*

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

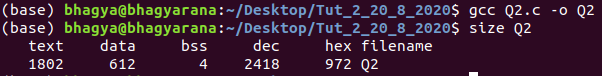
*return* n + sum(n - 1);

*else*

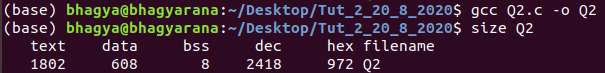
*return* n;

}

Q2 Initialized Code Size:



Q2 Uninitialized Code Size (Uncommenting Second Line):



Reason: We can clearly observe a difference of 4 bytes due to the fact that:

*Initialized Static Variable* is stored in **data segment** and *Uninitialized Static Variable* is stored in **BSS Segment**. Therefore, *Static Variables are stored data Segment or BSS depending on whether they are initialized or not.*

3.) When variable “number” is declared as an Extern variable

Q3 “Extfile.h”

int number = 1000;*// Intialised External Variable*

*//int number; // Unintialized External Variable*

Q3 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

*#include* "ExtFile.h"

int sum(int n);

int main()

{

*// "number" declared  in "ExtFile.h"*

*// "number" declared & intialised in "ExtFile.h"*

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

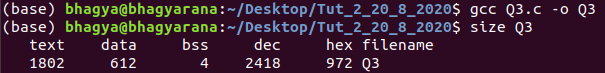
*return* n + sum(n - 1);

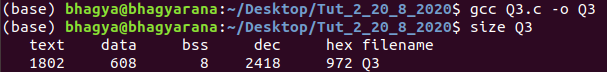
*else*

*return* n;

}

Q3 Initialized Code Size:



Q3 Uninitialized Code Size (Uncommenting Second Line in header file):

Reason: We can clearly observe a difference of 4 bytes due to the fact that:

*Initialized Extern Variable* is stored in **data segment** and *Uninitialized Extern Variable* is stored in **BSS Segment**. Extern Variable is Stored in “ExtFile.h” header file. Therefore, *Extern Variables are stored data Segment or BSS depending on whether they are initialized or not.*

4.) When variable “number” is declared as a Constant variable

Q4 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

int sum(int n);

*//When variable “number” is declared as a Constant variable*

int main()

{

    const int number = 100; *// number is declared as constant int*

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

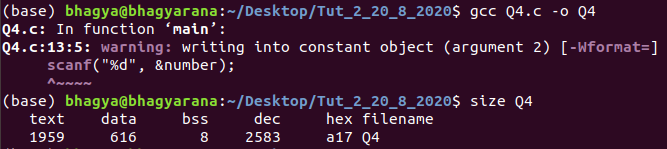
*return* n + sum(n - 1);

*else*

*return* n;

}

Q4 Initialized Code Size:



Reason: We can clearly observe Rise in 4 bytes in “data” Segment as compared to Initialized Code Size (i.e. 612). Therefore, *constant are stored in Stack Section of Data Segment.*

5.) When variable “number” is declared as an Auto variable

Q5 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

int sum(int n);

*//When variable “number” is declared as an Auto variable*

int main()

{

    auto number = 0; *// auto "number"*

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

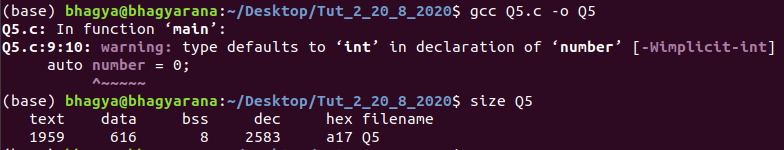
*// sum() function calls itself*

*return* n + sum(n - 1);

*else*

*return* n;

}

Q5 Initialized Code Size:

Reason: We can clearly observe Rise in 4 bytes in “data” Segment as compared to Initialized Code Size (i.e. 612). Therefore, “*auto” are stored in Stack Section of Data Segment.*

6.) When variable “number” is declared as a Register variable

Q6 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

int sum(int n);

*// When variable “number” is declared as a Register variable*

int main()

{

    int number = 0; *//itialised register*

*//int number;   //unitialised register*

    register int \*num\_add = &number;

    int result;

    printf("Enter a positive integer: ");

    scanf("%d", num\_add);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

    int p = n;

*if* (n != 0)

*// sum() function calls itself*

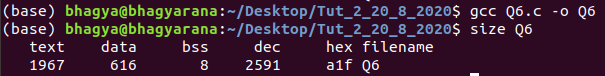
*return* n + sum(n - 1);

*else*

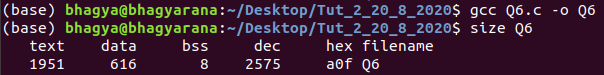
*return* n;

}

Q6 Initialized Code Size:



Q6 Uninitialized Code Size:



Reason: We can clearly observe Rise in 4 bytes in “data” Segment as compared to Initialized Code Size (i.e. 612). Therefore, “*register” are stored in Stack Section of Data Segment and Initializing does not have any effect in memory.*

7.) When variable “p” is declared as an Auto variable

Q7 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

int sum(int n);

int main()

{

    int number, result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

*// When variable “p” is declared as an Auto variable*

    auto p = n;

*if* (n != 0)

*// sum() function calls itself*

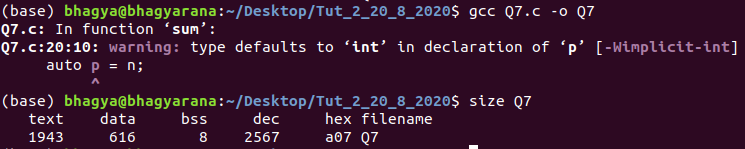
*return* n + sum(n - 1);

*else*

*return* n;

}

Q7 Initialized Code Size:



Reason: We can clearly observe Rise in 4 bytes in “data” Segment as compared to Initialized Code Size (i.e. 612). Therefore, “*auto” are stored in Stack Section of Data Segment.*

8.) When variable “p” is declared as a Static variable

Q8 Initialized Code:

*// C Code : Sum of n Natural Numbers Using Recursion*

*#include* <stdio.h>

int sum(int n);

int main()

{

    int number, result;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);

*return* 0;

}

int sum(int n)

{

*// When variable “p” is declared as an Static variable*

    static int p;

    p = n;

*if* (n != 0)

*// sum() function calls itself*

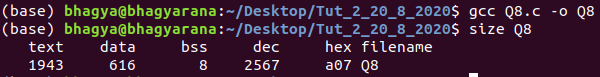
*return* n + sum(n - 1);

*else*

*return* n;

}

Q8 Initialized Code Size:



Reason: We can clearly observe Rise in 4 bytes in “data” Segment as compared to Initialized Code Size (i.e. 612). Therefore, *Static Variables are stored data Segment since they are initialized with value of n.*

Submitted By:

Roll Number: **U19CS012** (*D-12*)

Name: *Bhagya Rana*