**Lab #1: Numbering System**

Question #1/b

**101101012**= **18110**

n = 1 \* 20 + 1 \* 22 + 1 \* 24 + 1 \* 25 + 1 \* 27

 = 1 + 4 + 16 + 32 + 128

 = 181

Question #2/e

 ?

**10112 = 1110****√**

10112 = 1 \* 20 + 1 \* 21 + 1 \* 23

 = 1 + 2 + 8

 = 1110

Question #2/h

 ?

**11012 > 10102****√**

?

1 \* 20 + 1 \* 22 + 1 \* 23 > 1 \* 21 + 1 \* 23

 ?

1 + 4 + 8 > 2 + 8

1310 > 1010

Question #3/b

**3310**= **1000012**

33 / 2 = 16 R of **1** (rightmost digit)

 16 / 2 = 8 R of **0**

 8 / 2 = 4 R of **0**

 4 / 2 = 2 R of **0**

 2 / 2 = 1 R of **0**

 1 / 2 = 0 R of **1** (leftmost digit)

Question #4/d

**47.87510**= **101111.1112**

47 / 2 = 23 R of **1** (rightmost digit)

 23 / 2 = 11 R of **1**

 11 / 2 = 5 R of **1**

 5 / 2 = 2 R of **1**

 2 / 2 = 1 R of **0**

 1 / 2 = 0 R of **1** (leftmost digit)

0.875 \* 2 = **1** + 0.75 (leftmost digit)

0.75 \* 2 = **1** + 0.5

0.5 \* 2 = **1** + 0.0 (rightmost digit)

Question #5/d

**101111.1112** = (1.01111111 \* 25)2

Power = 5 + 127 = 132

|  |
| --- |
|  |
| 0 | 1000010 | 001111111000000000000000 |

423F800016

Question #6/c

**0.410**= **0.01102**

0.4 \* 2 = **0** + 0.8 (leftmost digit)

0.8 \* 2 = **1** + 0.6

0.6 \* 2 = **1** + 0.2

0.2 \* 2 = **0** + 0.4 (rightmost digit)

0.4

Question #7/a

**010011102 +** **001111002 = 100010102**

 **11111**

 01001110

 + 00111100

 **10001010**

Question #8/1

**-1610** = **111100002**

1’s Comp.

16**10** = 000100002 ------🡪 111011112

 +

 1

 11110000**2**

Question #8/5

**2610** = **000110102**

Question #9/d

**000001112 -** **3510**

1’s Comp.

35 = 001000112 ------🡪 110111002

 +

 1

 11011101**2**

 **11111**

00000111

 + 11011101

 **11100100**

Question #10/c

**DE016 = 355210**

 = 14 \* 161 + 13 \* 162

 = 224 + 3328

 = 3552

Question #10/g

**70.78 = 56.87510**

 = 7 \* 81 + 7 \* 8-1

 = 56 + 0.875

 = 56.875

Question #11

Ex. Ahmad

|  |
| --- |
| Parity bit |
| A | 1 | 1000001 | = C116 |

|  |
| --- |
| Parity bit |
| h | 0 | 1101000 | = 6816 |

|  |
| --- |
| Parity bit |
| m | 0 | 1101101 | = 6D16 |

|  |
| --- |
| Parity bit |
| a | 0 | 1100001 | = 6116 |

|  |
| --- |
| Parity bit |
| d | 0 | 1100100 | = 6416 |

|  |
| --- |
| C1 |
| 68 |
| 6D |
| 61 |
| 64 |

**Memory**

**Lab #2: Design computer algorithms**

Question #1

**Pseudocode:**

1 Set total to zero

2 Set counter to one

3 Input the first grade

4 While the users has not as yet entered -1

5 Add the grade into the total

6 Add one to the counter

7 Input the next grade (possibly the -1)

ENDWhile

8 IF the counter is not equal to zero THEN

9 Set the average to the total divided by counter

10 Print the average.

11 ELSE

12 Print “ No grades were entered”

ENDIF

**END**

Question #2

**Pseudocode:**

1 Set passes to zero

2 Set failures to zero

3 Set counter to one

4 Input the first grade

5 While the users has not as yet entered -1

6 IF the student passed THEN

7 Add one to passed

8 ELSE

9 Add one to failures

 ENDIF

10 Add one to counter

11 Input the next grade (possibly the -1)

 ENDWhile

12 Print “ The number of passes is “ passed

13 Print “ The number of failures is “ failures

14 Print “ The failure percentage is “failures/counter \* 100%

 **END**

Question #3

**Pseudocode:**

1. Set num\_items to zero
2. Set cost to zero
3. Input the price
4. WHILE price is NOT –1
5. cost = cost + price
6. add 1 to num\_items
7. get price

ENDWhile

1. tax = cost \* 0.075
2. total\_cost = cost +tax
3. display num\_items, cost, tax, total\_cost

**END**

Question #4

**Pseudocode:**

1. Input the first operand
2. Input the second operand
3. Input the operation
4. Case operation of
5. ‘+’ : result = first operand + second operand
6. ‘-‘ : result = first operand - second operand
7. ‘\*’ : result = first operand \* second operand
8. ‘%’ : result = first operand Mod second operand

ENDCase

1. Print the result

**END**

Question #5

**Pseudocode:**

1. Set total to zero
2. Set counter to one
3. Set factorial to one
4. While Counter is less than or equal to 8
5. factorial = factorial \* counter
6. total = total + factorial
7. Add one to counter

ENDWhile

1. Print the total

**END**

Question #6

**Pseudocode:**

1. Set counter to one
2. Input the number of integers
3. Input the first integer value
4. maximum = the first integer value
5. minimum = the first integer value
6. While counter is less than the number of integers
7. Input the next integer value
8. IF maximum is less than the next integer value THEN
9. maximum = the next integer value

 ENDIF

1. IF minimum is greater than the next integer value THEN IF maximum less than the next integer value THEN
2. minimum = the next integer value

 ENDIF

1. Add one to counter

ENDWhile

1. Print “ The Maximum number is “maximum
2. Print “ The Minimum number is “minimum

**END**

**Lab #3: C Building Blocks**

Question #1

/\*Write a program which shows the function of each escape sequence character.\*/

#include <stdio.h>

main()

{

 printf("alert ring bell rings like \a\a\a\a\a\a\a\a\ n");

 printf("the tab is inserted like \t this\n");

 printf("the new line like \n this\n");

 printf("the carriage return like \r this\n");

 printf("the backspace like\b this\n");

 printf("the double quotation like welcome to \"C\"\n");

 printf("the back slash like welcome to C\\C++\n");

 return 0;

}

Question #2

1. z = (4.2\*(x+y)\*5/z – 0.52\*x/(y+z))/((x+y)\*(x+y));
2. x = a \* a + 2 \* a \* b + b \* b;

Question #3

**Output:**

0.000000

 0.555556

Question #4

**Output:**

6 6

Question #5

int a = 5, b = 10, c;

c = (a > b) || (a < b); // 1

c = (a > b) && (a < b); // 0

c = (a != b) + (a < b); // 2

c = a + 5 != b \* 2 – 5; // 1

**Lab #4: Functions in C-Language programming**

Question #1

/\* Write a program to take a depth (in kilometers) inside the earth as input data; compute and display the temperature at this depth in degrees Celsius and degrees Fahrenheit, The relevant formulas are Celsius = 10 (depth) + 20

Fahrenheit = 1.8 ( Celsius) + 32

Include two functions in your program. Function Celsius\_at\_depth should compute and return the Celsius temperature at a depth measured in kilometers. Function Fahrenheit should convert a Celsius temperature to Fahrenheit.\*/

#include <stdio.h>

float celsius\_at\_depth(float);

float fehrenheit(float);

main()

{

 float depth,celsius;

 printf("Enter the depth inside the earth:\n");

 scanf("%f",&depth);

 celsius = celsius\_at\_depth(depth);

printf("The temperature at this depth in degree Celsius is %0.2f\n",celsius);

printf("The temperature at this depth in degree Fehrenheit is %0.2f\n",fehrenheit(celsius));

return 0;

}

float celsius\_at\_depth(float depth)

{

 return ( 10 \* depth + 20);

}

float fehrenheit(float celsius)

{

 return ( 1.8 \* celsius + 32);

}

Question #2

/\* Write a program to find

a. Surface area (A=4.r2)

b. volume(v=4/3\*(3.14) 3)

of a sphere using functions write a function for finding powers of radius. \*/

#include <stdio.h>

float surface\_area(float);

float volume(float);

float power2(float);

float power3(float);

main()

{

 float radius;

 printf("Enter the radius of the sphere:\n");

 scanf("%f",&radius);

 printf("The surface area is %0.2f\n",surface\_area(radius));

 printf("The volume is %0.2f\n",volume(radius));

return 0;

}

float surface\_area(float radius)

{

 return ( 4 \* 3.14 \* power2(radius));

}

float volume(float radius)

{

 return ( 4.0/3 \* 3.14 \* power3(radius));

}

float power2(float radius)

{

 return ( radius \* radius);

}

float power3(float radius)

{

 return ( radius \* radius \* radius);

}

Question #3

**Solution 1**

/\* Write a program that takes a positive number with a fractional part and rounds it to two decimal places. For example, 32.4851 would round to 32.49, and 32.4431 would round to 32.44 \*/

#include <stdio.h>

main()

{

 float number;

 printf("Enter a number:\n");

 scanf("%f",&number);

 printf("The round to two decimal places is %0.2f\n",number);

return 0;

}

**Solution 2**

/\* Write a program that takes a positive number with a fractional part and rounds it to two decimal places. For example, 32.4851 would round to 32.49, and 32.4431 would round to 32.44 \*/

#include <stdio.h>

float round(float);

main()

{

 float number;

 printf("Enter a number:\n");

 scanf("%f",&number);

printf("The round to two decimal places is %0.2f\n",

 round(number));

return 0;

}

float round(float number)

{

 int num = number \* 100;

 if ((number \* 100 - num) >= 0.5)

 ++num;

 return num/100.0;

}

**Lab #5: Decision making the if, if-else, Switch case, and conditional operator**

Question #1

/\* Write a program which takes three sides a, b and c of a triangle as input and calculates its area if these conditions are satisfied a+b>c, b+c>a, and a+c>b

(Help a= √ s(s-a)(s-b)(s-c) , where s=(a+b+c)/2 ) \*/

#include <stdio.h>

#include <math.h>

float area(float, float, float);

main()

{

 float a,b,c;

 printf("Enter the three triangle sides:\n");

 scanf("%f %f %f",&a,&b,&c);

 if ( (a+b>c) && (b+c>a) && (a+c>b))

 printf("%0.2f", area(a,b,c));

 else

 printf("Invalid values");

return 0;

}

float area(float a,float b, float c)

{

 float s = (a+b+c)/ 2;

 return (sqrt(s\*(s-a)\*(s-b)\*(s-c)));

}

Question #2

/\* Write a program that inputs an integer – determines if it is even or odd. \*/

#include <stdio.h>

int is\_even(int);

main()

{

 int number;

 printf("Enter a number:\n");

 scanf("%d",&number);

 if (is\_even(number))

 printf("Even number");

 else

 printf("Odd number");

return 0;

}

int is\_even(int num)

{

 if (num % 2 == 0)

 return 1;

 else

 return 0;

}

Question #3

/\* Write a program which takes a character as input and checks whether it is a vowel or consonant.\*/

#include <stdio.h>

int is\_vowel(char);

main()

{

 char ch;

 printf("Enter a character:\n");

 scanf("%c",&ch);

 if (is\_vowel(ch))

 printf("is vowel character");

 else

 printf("is consonant cahracter");

return 0;

}

int is\_vowel(char ch)

{

 switch (ch)

 {

 case 'a':case 'A':

 case 'e':case 'E':

 case 'i':case 'I':

 case 'o':case 'O':

 case 'u':case 'U': return 1;

 }

 return 0;

}

Question #4

/\* Write a program which takes a character as input and check

 whether it is a vowel, consonant, or special character.\*/

#include <stdio.h>

int check\_character(char);

main()

{

 char ch;

 printf("Enter a character:\n");

 scanf("%c",&ch);

 if (check\_character(ch) == 1)

 printf("is consonant cahracter");

 else

 if (check\_character(ch) == 2)

 printf("is vowel character");

 else

 printf("is special cahracter");

 return 0;

}

int check\_character(char ch)

{

 int status; // 1: vowel 2: consonant 3:other

 if ((ch >= 'A' && ch <= 'Z') ||(ch >= 'a' && ch <= 'z'))

 switch (ch)

 {

 case 'a':case 'A':

 case 'e':case 'E':

 case 'i':case 'I':

 case 'o':case 'O':

 case 'u':case 'U': status = 2;

 break;

 default : status = 1;

 }

 else

 status = 3;

 return status;

}

Question #5

/\* Write a program to make a simple calculator which should be able to do +,-,\*,/,% Operations. \*/

#include <stdio.h>

void calculate(char,int,int);

main()

{

 char op;

 int number1, number2;

 printf("Enter the operation:\n");

 scanf("%c",&op);

 printf("Enter the first and the second integers:\n");

 scanf("%d %d",&number1, &number2);

 calculate(op,number1,number2);

return 0;

}

void calculate(char op,int num1,int num2)

{

 switch (op)

 {

 case '+':printf("%d",num1 + num2);

 break;

 case '-':printf("%d",num1 - num2);

 break;

 case '\*':printf("%d",num1 \* num2);

 break;

 case '/':printf("%d",num1 / num2);

 break;

 case '%':printf("%d",num1 % num2);

 break;

 default : printf("Invalid operation");

 }

}

Question #6

/\* Write a program which takes 5 integers as input and prints the largest one. \*/

#include <stdio.h>

int largest\_number(int,int,int,int,int);

main()

{

 int number1,number2, number3, number4, number5;

 printf("Enter five numbers:\n");

 scanf("%d %d %d %d %d",&number1, &number2,&number3,

 &number4,&number5);

 printf("The largest number is %d",

largest\_number(number1,number2,number3,number4,number5));

return 0;

}

int largest\_number(int num1,int num2,int num3,int num4,int num5)

{

 int max = num1;

 if (max < num2)

 max = num2;

 if (max < num3)

 max = num3;

 if (max < num4)

 max = num4;

 if (max < num5)

 max = num5;

 return max;

}

**Lab #6: Looping constructs in C-Language and nested loops**

Question #1

i.

**Output:**

1

2

3

4

5

6

7

8

9

10

ii.

**Output:**

 Infinite loop

Question #2

/\* Write a program to generate a series of the first positive 50 even numbers. \*/

#include <stdio.h>

main()

{

 int i;

 for( i=0; i<=50 ; i+=2)

 {

 printf("%d\t", i);

 }

 return 0;

}

Question #3

/\* Write a program to generate a multiplication tables from 2 to 20 with first 10 terms. \*/

#include <stdio.h>

main()

{

 int i,j;

 for( i=2; i<=20 ; i++)

 {

 printf("Multiplication table for %d is\n",i);

 for( j=1; j<=10; j++)

 printf("\t%d \* %d = %d\n", i,j,i\*j);

 }

 return 0;

}

Question #4

**Solution 1**

/\* Write a program segment, which may be used to input a sentence. Terminate when Enter key is pressed. (Use for loop). \*/

#include <stdio.h>

main()

{

 char ch;

 for( ;ch != '\n'; )

 {

 scanf("%c",&ch);

 }

 return 0;

}

**Solution 2**

/\* Write a program segment, which may be used to input a sentence. Terminate when Enter key is pressed. (Use for while). \*/

#include <stdio.h>

main()

{

 char ch;

 while(ch != '\n')

 {

 scanf("%c",&ch);

 }

 return 0;

}

Question #5

**Output:**

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Question #6

/\* Write a program to print a series of the first positive 50 prime numbers.\*/

#include <stdio.h>

main()

{

 int i,j,flage;

 printf("the first positive 50 prime numbers are\n");

 printf("2\n");

 for ( i=3; i<=50;i+=2)

 {

 flage = 1;

 for (j=2; j<=i/2;j++)

 {

 if ( i%j == 0)

 flage = 0;

 }

 if (flage)

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

 }

 return 0;

}

Question #7

#include <stdio.h>

main()

{

 int i,j;

 for(i= 0 ; i<10; i++)

 {

 for(j = 1; j<= 10-i; j++)

 printf(" ");

 for(j=1; j<= i\*2 + 1; j++)

 printf("%d",i);

 printf("\n");

 }

 return 0;

}

Question #8

/\* Write a program to print the following pattern up to Z only

A B C D

E F G H

I J K L

 \*/

#include <stdio.h>

main()

{

 int i;

 for ( i=1; i<= 26; i++)

 {

 printf("%c\t",i+64);

 if ( i % 4 == 0)

 printf("\n");

 }

 return 0;

}

**Lab #7: Modular Programming**

Question #1

/\* Write a program to dispense change. The user enters the amount paid and the amount due. The program determines how many dollars, quarters, dimes, nickels, and pennies should be given as change. Write a function with four output parameters that determines the quantity of each kind of coin.\*/

#include <stdio.h>

#include <math.h>

main()

{

 int dollars, quarters, dimes, nickels, pennies, changes;

 float total, paid;

 printf("Enter the total amount ");

 scanf("%f",&total);

 printf("Enter the paid amount ");

 scanf("%f",&paid);

 changes = paid - total;

 if ( paid - total > 0)

 {

 dollars = floor(paid - total) ;

 changes = (paid - total - floor(paid - total)) \* 100;

 quarters = changes / 25;

 changes = changes % 25;

 dimes = changes / 10;

 changes = changes % 10 ;

 nickels = changes / 5;

 pennies = changes % 5 ;

 printf("%d dollars , %d quarters, %d dimes, %d nickels,

 and %d pennies\n", dollars, quarters, dimes,

 nickels, pennies);

 }

 else

 printf("not enough amount");

 return 0;

}

Question #2

/\* The square root of a number N can be approximated by repeated calculation using the formula

NG = 0.5 ( LG + N/LG )

where NG stands for next guess and LG stands for last guess. Write a function that calculates the square root of a number using this method. The loop should be repeated until the difference is less than 0.005. Use an initial guess of 1.0.

Write a driver function and test your square root function for the numbers 4, 120.5, 88, 36.01, 10,000, and 0.25. \*/

#include <stdio.h>

#include <math.h>

float squareRoot(float);

main()

{

 float num;

 printf("Enter a number\n");

 scanf("%f",&num);

 printf("Square root of %0.2f is %0.2f",num,squareRoot(num));

 return 0;

}

float squareRoot(float n)

{

 float lg = 1.0;

 while ( fabs(lg- sqrt(n)) > 0.005)

 {

 lg = 0.5 \* (lg + n/lg);

 }

 return lg;

}

Question #3

#include <stdio.h>

 float charge(float);

main()

{

 int Customer\_Number;

 float Kilowatt\_hours;

 printf("Enter the Customer Number and the kilowatt/hours

used. 0 to end input\n");

 scanf("%d %f",&Customer\_Number,&Kilowatt\_hours);

 while (Customer\_Number != 0)

 {

 printf("%d\t%0.2f\t%0.2f\n",Customer\_Number,Kilowatt\_hours , charge(Kilowatt\_hours));

 printf("?\n");

 scanf("%d %f",&Customer\_Number,&Kilowatt\_hours);

 }

 return 0;

}

float charge (float n)

{

 float total = 0;

 if ( n > 1000)

 {

 total += (n - 1000) \* 0.05;

 n=1000;

 }

 if ( n > 600)

 {

 total += (n - 600) \* 0.06;

 n=600;

 }

 if ( n > 300)

 {

 total += (n - 300) \* 0.07;

 n=300;

 }

 if ( n > 0)

 total += n \* 0.08;

 return total;

}

**Lab #8: Arrays in C (one dimensional)**

Question #1

/\* Write a program that declare two arrays of integers. Then fill them and exchanges their values.\*/

#include <stdio.h>

float exchange (int [],int [],int);

main()

{

 const int size=10;

 int arr1[size], arr2[size], i;

 printf("Enter the first %d values\n",size);

 for ( i=0; i< size; i++)

 scanf("%d",&arr1[i]);

 printf("Enter the second %d values\n",size);

 for ( i=0; i< size; i++)

 scanf("%d",&arr2[i]);

 exchange(arr1,arr2,size);

 printf("The first %d values\n",size);

 for ( i=0; i< size; i++)

 printf("%d\n",arr1[i]);

 printf("The second %d values\n",size);

 for ( i=0; i< size; i++)

 printf("%d\n",arr2[i]);

 return 0;

}

float exchange (int array1[],int array2[],int n)

{

 int temp, i;

 for (i = 0; i<n; i++)

 {

 temp = array1[i];

 array1[i] = array2[i];

 array2[i] = temp;

 }

}

Question #2

/\* Write a program that takes 10 integers as input and prints the largest integer and its location in the array. \*/

#include <stdio.h>

int largest (int [],int & ,int);

main()

{

 const int size=10;

 int arr1[size],max, location=0, i;

 printf("Enter %d values\n",size);

 for ( i=0; i< size; i++)

 scanf("%d",&arr1[i]);

 max = largest(arr1,location,size);

 printf("The maximum value is %d\n",max);

 printf("The location of the maximum value is %d\n",location);

 return 0;

}

int largest (int array[],int &location, int n)

{

 int max=array[0], loc= 0, i;

 for (i = 1; i<n; i++)

 {

 if (max < array[i] )

 {

 max = array[i];

 location = i;

 }

 }

 return max;

}

Question #3

/\* Write a program which takes a string as input and then counts the total number of vowels in that string. \*/

#include <stdio.h>

int is\_vowel(char );

main()

{

 char ch;

 int vowelCount = 0;

 for( ;ch != '\n'; )

 {

 scanf("%c",&ch);

 vowelCount += is\_vowel(ch);

 }

 printf("The total number of vowels are %d",vowelCount);

 return 0;

}

int is\_vowel(char ch)

{

 switch (ch)

 {

 case 'a':case 'A':

 case 'e':case 'E':

 case 'i':case 'I':

 case 'o':case 'O':

 case 'u':case 'U': return 1;

 }

 return 0;

}

Question #4

/\* Write a program to sort an integer array in descending order. \*/

#include <stdio.h>

void sort (int [],int);

main()

{

 const int size=10;

 int arr1[size],i;

 printf("Enter %d values\n",size);

 for ( i=0; i< size; i++)

 scanf("%d",&arr1[i]);

 sort(arr1,size);

 printf("sorting an integer array in descending order\n");

 for ( i=0; i< size; i++)

 printf("%d\n",arr1[i]);

 return 0;

}

void sort (int array[],int n)

{

 int temp, j, i;

 for (i = 0; i<n-1; i++)

 {

 for ( j=i+1; j< n; j++)

 if (array[i] < array[j] )

 {

 temp = array[i];

 array[i] = array[j];

 array[j] = temp;

 }

 }

}

**Lab #9: Arrays in C (Multidimensional)and string functions.**

Question #1

/\* Write a program that adds up two 4x4 arrays and stores the sum in third array. \*/

#include <stdio.h>

void add (int [][4], int[][4], int [][4], int, int);

main()

{

 const int column=4, row= 4;

 int a[row][column], b[row][column], c[row][column], i, j;

 printf("Enter the elements of the first matrix\n");

 for ( i=0; i< row; i++)

 {

 for (j = 0; j < column; j++)

 {

 printf("a[%d][%d] = ",i,j);

 scanf("%d",&a[i][j]);

 }

 }

 printf("Enter the elements of the second matrix\n");

 for ( i=0; i< row; i++)

 {

 for (j = 0; j < column; j++)

 {

 printf("b[%d][%d] = ",i,j);

 scanf("%d",&b[i][j]);

 }

 }

 add(a,b,c,row,column);

 printf("c = a + b\nc = \n");

 for ( i=0; i< row; i++)

 {

 for (j = 0; j < column; j++)

 {

 printf("%d",c[i][j]);

 printf("\t");

 }

 printf("\n");

 }

 return 0;

}

void add (int a[][4], int b[][4], int c[][4],int row, int column)

{

 int j, i;

 for (i = 0; i<row; i++)

 {

 for ( j=0; j< column; j++)

 c[i][j] = a[i][j] + b[i][j] ;

 }

}

Question #2

/\* Write a program which takes names of five countries as input and prints them in alphabetical order. \*/

#include <stdio.h>

#include <string.h>

typedef char string[30];

void sort (string [],int);

main()

{

 const int size=5;

 string arr1[size];

 int i;

 printf("Enter %d countries\n",size);

 for ( i=0; i< size; i++)

 scanf("%s",arr1[i]);

 sort(arr1,size);

 printf("sorting a string array in alphabetical order\n");

 for ( i=0; i< size; i++)

 printf("%s\n",arr1[i]);

 return 0;

}

void sort (string st[],int n)

{

 int i,j;

 string temp;

 for (i = 0; i<n-1; i++)

 {

 for ( j=i+1; j< n; j++)

 if (strcmp(st[i],st[j])> 0 )

 {

 strcpy(temp,st[i]);

 strcpy(st[i],st[j]);

 strcpy(st[j],temp);

 }

 }

}

Question #3

/\* Write and test a function hydroxide that returns a 1 for true if its string argument ends in the substring OH. \*/

#include <stdio.h>

#include <string.h>

typedef char string[30];

int is\_hydroxide (string);

main()

{

 string st;

 printf("Enter a string\n");

 scanf("%s",st);

 if (is\_hydroxide(st))

 printf("The string is hydroxide\n");

 else

 printf("The string is not hydroxide\n");

 return 0;

}

int is\_hydroxide (string st)

{

 if ((st[strlen(st)-2] == 'H') && (st[strlen(st)-1] == 'O'))

 return 1;

 return 0;

}

Question #4

/\* Write a program that takes data one line at a time and reverses the words of the line. For example,

Input : birds and bees

Output : bees and birds \*/

#include <stdio.h>

#include <string.h>

typedef char string[30];

void reverses (string, string);

main()

{

 string st, out\_st = "";

 printf("Enter a string\n");

 gets(st);

 reverses(st,out\_st);

 printf("The reverse string is %s",out\_st);

 return 0;

}

void reverses (string st, string out\_st)

{

 string temp = "";

 char \*p;

 p = strtok(st," ");

 while (p != NULL)

 {

 strcpy(temp,p);

 strcat(strcat(temp," "),out\_st);

 strcpy(out\_st,temp);

 p = strtok(NULL," ");

 }

}

**Lab #10: Structures**

Question #1

/\* Write a program to maintain the library record for 100 books with book name, author’s name, and edition, year of publishing and price of the book. \*/

#include <stdio.h>

 struct book

{

 char name[20];

 char author[15];

 short edition;

 int year;

 float price;

};

main()

{

 const int size = 100;

 book lib[size];

 int i;

 printf("Enter a book name,author,edition,year of publishing

 and price of 100 books\n");

 for ( i = 0; i< size; i++)

 {

 scanf("%s %s %d %d %f",&lib[i].name,&lib[i].author,

 &lib[i].edition, &lib[i].year, &lib[i].price);

 }

 printf("Book Name \t author\t edition \t year \t price\n");

 for ( i = 0; i< size; i++)

 {

 printf("%s\t %s\t %d\t %d\t %0.2f\n",lib[i].name,

lib[i].author, lib[i].edition, lib[i].year, lib[i].price );

 }

 return 0;

}

Question #2

/\* Write a program to make a tabulation sheet for a class of 50 students with their names, seat no's, marks, percentages and grades. \*/

#include <stdio.h>

struct class\_sheet

{

 char name[10];

 unsigned seat\_no;

 int marks;

 float percentages;

 char grades;

};

main()

{

 const int size = 50;

 class\_sheet sheet[size];

 int i;

 printf("Enter the student name,seat no's, marks, percentages

 and grades of %d students\n",size);

 for ( i = 0; i< size; i++)

 {

 scanf("%s %d %d %f %c",&sheet[i].name,&sheet[i].seat\_no,

&sheet[i].marks, &sheet[i].percentages, &sheet[i].grades );

 printf("?\n");

 }

 printf("student Name \tseat No.\tmarks \tpercentages

\tgrades\n");

 for ( i = 0; i< size; i++)

 {

 printf("%s\t\t %d\t\t %d\t %0.2f\t\t %c\n",

 sheet[i].name,sheet[i].seat\_no, sheet[i].marks,

 sheet[i].percentages, sheet[i].grades);

 }

 return 0;

}

Question #3

/\* Define a structure to represent a complex number in rectangular format.

Complex numbers have the form

realPart + imaginaryPart \* i \*/

struct complexNumber

{

 float realPart;

 float imaginaryPart;

};

Question #4

/\* Consider the following structure definition:

 struct Student

 {

 char name[10];

 int section;

 float grade;

 };

Write a program that declare a list(struct Student)with size > 0. Your Program will do the following:

1.Raise all the grades in the class 5% to a maximum of 100.

2.Give everyone in the class whose name starts with “Al” a grade of 100.

3.Sort the list by grades, highest to lowest. \*/

#include <stdio.h>

#include <string.h>

struct Student

{

 char name[20];

 int section;

 float grade;

};

void raiseGrades(Student [], int, const int);

void grade100(Student [], int, const char [], int);

void sort(Student [], int);

main()

{

 const int maxSize = 50;

 int actualSize, i;

 Student stud[maxSize];

 printf("Enter the number of students\n");

 scanf("%d",&actualSize);

 if (actualSize > 0)

 {

 for ( i = 0; i< actualSize; i++)

 {

 printf("Enter the student name, section, and grades

 for student No. %d\n",i+1);

 scanf("%s %d %f",stud[i].name,&stud[i].section,

 &stud[i].grade);

 }

 raiseGrades(stud, actualSize, 100);

 grade100(stud , actualSize, "Al", 100);

 sort(stud, actualSize);

 printf("Student Name\tsection \tgrades\n");

 for ( i = 0; i< actualSize; i++)

 {

 printf("%s\t\t %d\t\t%0.2f\n",

 stud[i].name,stud[i].section, stud[i].grade);

 }

 }

 else

 printf("The size must be greater than zero");

 return 0;

}

void raiseGrades(Student st[], int size, const int maxGrade)

{

 int i;

 for ( i= 0; i < size ; i++)

 st[i].grade = (st[i].grade \* 1.05 > maxGrade) ? maxGrade

 : st[i].grade \* 1.05;

}

void grade100(Student st[], int size, const char token[], int maxGrade)

{

 int i;

 char s[3]= " ";

 for ( i= 0; i < size ; i++)

 if (strcmp(strncpy(s,st[i].name,2),token) == 0)

 st[i].grade = maxGrade;

}

void sort(Student st[], int size)

{

 Student temp;

 int i, j;

 for (i = 0; i< size-1; i++)

 for( j=i+1; j < size; j++)

 if (st[i].grade < st[j].grade)

 {

 temp = st[i];

 st[i] = st[j];

 st[j] = temp;

 }

}

**Lab #11: Filing in C-Language**

Question #1

/\* What will be the output of the given program \*/

#include <stdio.h>

main( )

See the modification her

{

 FILE\*fpt;

 char c;

 fpt = fopen ("star.dat","**w**"); // a new file is made

 do

 {

 putc((c= getchar( ) )**,** fpt );

 }while(c!= '\n' ); // or '\r'

See the modification her

 fclose( fpt );

 return 0;

}

Question #2

/\* Write a program to store strings in a file. \*/

#include <stdio.h>

main( )

{

 FILE\*fpt;

 char st[10];

 short num, i;

 fpt = fopen ( "output.dat","w"); // a new file is made

 printf("Ente the number of strings\n");

 scanf("%d",&num);

 for(i=0; i<num; i++)

 {

 printf("?\n");

 scanf("%s",st);

 fprintf(fpt,"%s\n",st);

 }

 fclose( fpt );

 return 0;

}

Question #3

/\* Write a program segment that writes an array to a file. \*/

#include <stdio.h>

struct Student

{

 char name[10];

 int grade;

};

main( )

{

 const int size = 5;

 FILE\*fpt;

 Student stud[size];

 short num, i;

 fpt = fopen ("output.dat","w"); // a new file is made

 printf("Ente the number of students\n");

 scanf("%d",&num);

 for(i=0; i<num; i++)

 {

 printf("The name and grade fro student No. %d\n",i+1);

 scanf("%s %d",stud[i].name, &stud[i].grade);

 fprintf(fpt,"%s\t%d\n",stud[i].name, stud[i].grade);

 }

 fclose( fpt );

 return 0;

}

**Lab #12: Pointers in C-Language**

Question #1

/\* Write down the number of bytes allocated for the following pointer variables: \*/

int \*x; // 4 bytes //

char \*y; // 4 bytes

float \*z; // 4 bytes

Question #2

/\* Determine the output of the following program:

#include <stdio.h>

int main()

{

 int q=2;

 int \*p;

 p=&q;

 \*p=100;

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

 printf("%p\n",p);

 printf("%d\n",\*p);

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

 printf("%p\n",&q);

 printf("%p\n",&p);

 return 0;

} \*/

**Output:**

100

0022FF74

100

100

0022FF74

0022FF70

Question #3

/\* Determine the output of the following program:

#include <stdio.h>

int main()

{

 int x=3,y=4,z=6;

 int \*p1,\*p2,\*p3;

 p1=&x;

 p2=&y;

 p3=&z;

 \*p1=\*p2+\*p3;

 \*p1++;

 \*p2--;

 \*p1=(\*p2)\*(\*p3);

 \*p2=(\*p2)\*(\*p1);

 x=y+z;

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

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

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

 printf("%d\n",\*p1);

 printf("%d\n",\*p2);

 printf("%d\n",\*p3);

 return 0;

}\*/

**Output:**

220

4

216

36

216

216

Question #4

/\* Write a program which adds two arrays using their pointers. \*/

#include <stdio.h>

int main()

{

 const int size = 5;

 int x[size], y[size];

 int \*p1,\*p2;

 p1=x;

 p2=y;

 printf("Enter %d elements for the first array\n",size);

 for ( int i = 0; i< size; i++)

 scanf("%d",&x[i]);

 printf("Enter %d elements for the second array\n",size);

 for ( int i = 0; i< size; i++)

 scanf("%d",&y[i]);

 printf("The result is\n");

 for ( int i = 0; i < size; i++)

 {

 printf("%d\n",\*p1 + \*p2);

 p1++;

 p2++;

 }

 return 0;

}

**Lab #13: Pointers with arrays and function.**

Question #1

/\* Write a program to pass an integer array of 10 elements to a function which returns the same array after sorting it in descending order. Print the array. \*/

#include <stdio.h>

void sort (int \*,int);

void print (int \*,int);

main()

{

 const int size=10;

 int arr1[size],i;

 printf("Enter %d values\n",size);

 for ( i=0; i< size; i++)

 scanf("%d",&arr1[i]);

 sort(arr1,size);

 printf("sorting an integer array in descending order\n");

 print(arr1,size);

 return 0;

}

void sort (int \*a,int n)

{

 int temp, j, i;

 for (i = 0; i<n-1; i++)

 {

 for ( j=i+1; j< n; j++)

 if (a[i] < a[j] )

 {

 temp = a[i];

 a[i] = a[j];

 a[j] = temp;

 }

 }

}

void print (int \*a, int n)

{

 for ( int i=0; i< n; i++,a++)

 printf("%d\n",\*a);

}

Question #2

/\* Write a program which passes a string to a function and the function changes its case without using any library function. \*/

#include <stdio.h>

void changesCase (const char \*, char \*);

main()

{

 char st[30], out\_st[30];

 printf("Enter a string\n");

 gets(st);

 changesCase(st,out\_st);

 printf("The new string is %s",out\_st);

 return 0;

}

void changesCase (const char \*st, char \*out\_st)

{

 for(; (\*out\_st = \*st) != '\0'; st++, out\_st++)

 {

 if ((\*st >= 'a') && (\*st <= 'z'))

 \*out\_st = \*st - 32;

 else

 if ((\*st >= 'A') && (\*st <= 'Z'))

 \*out\_st = \*st + 32;

 }

}