Homework Hustlers: https://discord.gg/aJ55rZBV

- Wizard.

LABREPORTS

1. Write a C program to find cube of a number using function.

```
~
#include <stdio.h>
#define CUBE(x) x*x*x
int main(){
    printf("Cube: %d", CUBE(3));
    return 0;
}
```

```
~

Cube: 27
```

2. Write a program in C to swap two numbers using function.

```
~ Swapping Without Temporary Variable Using XOR

void swap( int *a, int *b){
    *a = (*a ^= *b,*b ^= *a, *a ^ *b);
}

int main(){

    int a,b;
    printf("Enter 2 numbers: (a,b): ");
    scanf("%d,%d",&a,&b);
    printf("a,b: %d, %d\n",a,b);
    swap(&a,&b);
    printf("a,b: %d, %d",a,b);

}
```

```
~ Outputs:

[wizard@archlinux w2]$ ./a.out
Enter 2 numbers: (a,b): 1,2
a,b: 1, 2
a,b: 2, 1
[wizard@archlinux w2]$
```

Explanation: Primeagen's Magic Swap

3. Write a void function which finds and prints the midpoint coordinates of a line. The function should take in four parameters (x1, y1, x2 and y2).

```
~

# include <stdio.h>
void findMidpoint(double a, double b, double c , double d){
    printf("Midpoints: (%.2f, %.2f)\n",((a+b)/2),((c+d)/2));
}

int main(){
    findMidpoint(1,3,5,7);
    return 0;
}
```

```
~

[wizard@archlinux w2]$ ./a.out
Midpoints: (2.00, 6.00)
[wizard@archlinux w2]$
```

4. Write a program in C to check Armstrong and perfect numbers using the function.

```
~

#include <stdio.h>
#include <math.h> // needs the gcc main.c -lm flag

void isArmstrong(int n){
    int sum = 0;
    for (int tmp=n, ndigits = (int)log10(n)+1; tmp; tmp / = 10) {
        sum += pow(tmp%10,ndigits);
    }
    printf(n == sum? "Armstrong\n":"Not Armstrong\n");
}

void isPerfect(int n){
    int sum = 0;
    for (int i=1;i<n;i++)
        sum += (n % i == 0)? i:0;
    printf(sum == n? "Perfect": "Not Perfect");

}

int main(){
    int n, sum = 0;
    printf("Enter a number: ");
    scanf("%d",&n);
    isArmstrong(n);
    isPerfect(n);
    return 0;
}
```

```
~

[wizard@archlinux w2]$ ./a.out
Enter a number: 6
Armstrong
Perfect
[wizard@archlinux w2]$ ./a.out
Enter a number: 28
Not Armstrong
Perfect
[wizard@archlinux w2]$
```

5. Write a function named “velocityCalc” which returns an appropriate value for the formula “v=u+at”.

Where v is the final velocity, u is the initial velocity, a is the acceleration and t is the time that has elapsed. Depending upon which variable is set to “NAN” when the function is called, your function should work it out and return the value. Make sure to take inputs from the user and let them decide which variable should be NAN, and the program should not work if more than one variable is NAN.

```
~

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

double calcVelocity(double v, double u, double a, double t) {
    if (isnan(v)) return u + a * t;
    if (isnan(u)) return v - a * t;
    if (isnan(a)) return (v - u) / t;
    if (isnan(t)) return (v - u) / a;
    return 0;
}

void parseInput(char *input, double *value) {
    char *token = strtok(input, ",");
    *value = (strcmp(token, "_") == 0) ? NAN : atof(token);
}

int main() {
    char input[50];
    double v, u, a, t;

    printf("Enter v,u,a,t (use _ for unknowns): ");
    scanf("%s", input);

    parseInput(input, &v);
    parseInput(NULL, &u);
    parseInput(NULL, &a);
    parseInput(NULL, &t);

    if (isnan(v) + isnan(u) + isnan(a) + isnan(t) != 1) {
        printf("Can't have more than 1 unknown.\n");
        return 1;
    }

    double result = calcVelocity(v, u, a, t);
    printf("Result: %f\n", result);
    return 0;
}
```

6. Write a void function named “equations” which solves simultaneous equations.

Your program will take six parameters. E.g. function(double a, double b, double c, double d, double e, double f){}. By solving simultaneous equations, you are finding where the two lines cross each other, so your function should print an x and y coordinate.

- ax+by=c ... (i)
- dx+ey=f ... (ii)
- a = number in front of x of equation one
- b = number in front of y of equation one
- c = constant of equation one
- d = number in front of x of equation two
- e = number in front of y of equation two

- f = constant of equation two

```
~

#include <stdio.h>

void equations(double a, double b, double c, double d, double e, double f) {
    double x = (e*c - b*f) / (a*e - b*d);
    double y = (a*f-d*c) / (a*e - b*d);
    printf("x: %f, y: %f",x,y);
}

int main(){
    equations(5.0,6.0,7.0,8.0,9.0,10.0);
}
```

```
~
[wizard@archlinux w2]$ ./a.out
x: -1.000000, y: 2.000000
[wizard@archlinux w2]$
```