Trabalho de Estrutura de Dados II

Nome: Lucas Sousa dos Reis
Matrícula: 20382356245

Professor: Enoque Alves

1d-arrays-in-c.c

```c
#include <stdio.h>

int main() {
    int tam, sum;

    scanf("%d", &tam);

    int array[tam];

    for (int i = 0; i < tam; i++) {
        scanf("%d", &array[i]);
    }

    sum = 0;

    for (int i = 0; i < tam; i++) {
        sum = sum + array[i];
    }

    printf("%d", sum);

    return 0;
}
```

array-reversal.c

```c
#include <stdio.h>

int main() {
    int tam;

    scanf("%d", &tam);

    int array[tam];

    for (int i = 0; i < tam; i++) {
        scanf("%d", &array[i]);
    }

    for (int i = tam - 1; i >= 0; i--) {
        printf("%d ", array[i]);
    }

    return 0;
}
```

boxes-through-a-tunnel.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_HEIGHT 41

struct box {
    int length;
    int width;
    int height;
};

typedef struct box box;

int get_volume(box b) {
    return b.length * b.height * b.width;
}

int is_lower_than_max_height(box b) {
    if (b.height < MAX_HEIGHT) {
        return 1;
    } else {
       return 0 ;
    }

}

int main()
{
    int n;
    scanf("%d", &n);
    box *boxes = malloc(n * sizeof(box));
    for (int i = 0; i < n; i++) {
        scanf("%d%d%d", &boxes[i].length, &boxes[i].width, &boxes[i].height);
    }
    for (int i = 0; i < n; i++) {
        if (is_lower_than_max_height(boxes[i])) {
            printf("%d\n", get_volume(boxes[i]));
        }
    }
    return 0;
}
```

calculate-the-nth-term.c

```c
#include <stdio.h>

int main() {
    int n;

    scanf("%d", &n);

    int array[n];

    for (int i = 0; i <= 2; i++) {
        scanf("%d", &array[i]);
    }

    for (int i = 3; i < n; i++) {
        array[i] = array[i - 1] + array[i - 2] + array[i - 3];
    }

    printf("%d", array[n - 1]);

    return 0;
}
```

conditional-statements-in-c.c

```c
#include <stdio.h>

int main() {
    int x;
    scanf("%d", &x);

    if (x == 0) {
        printf("zero");
    }

    else if (x == 1) {
        printf("one");
    }

    else if (x == 2) {
        printf("two");
    }

    else if (x == 3) {
        printf("three");
    }

    else if (x == 4) {
        printf("four");
    }

    else if (x == 5) {
        printf("five");
    }

    else if (x == 6) {
        printf("six");
    }

    else if (x == 7) {
        printf("seven");
    }

    else if (x == 8) {
        printf("eight");
    }

    else if (x == 9) {
        printf("nine");
    }

    else {
        printf("Greater than 9");
    }
```

```
    return 0;
}
```

functions-in-c.c

```c
#include <stdio.h>

int greatest_func(int a, int b, int c, int d) {
    int greatest;

    greatest = a;

    if (greatest < b) {
        greatest = b;
    }

    if (greatest < c) {
        greatest = c;
    }

    if (greatest < d) {
        greatest = d;
    }

    return greatest;

}

int main() {
    int greatest, a, b, c, d;

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

    greatest = greatest_func(a, b, c, d);

    printf("%d", greatest);

    return 0;
}
```

hello-world-in-c.c

```c
#include <stdio.h>

int main() {
    char x[30];

    gets(x);

    printf("Hello, World!\n");
    printf("%s", x);

    return 0;
}
```

playing-with-characters.c

```c
#include <stdio.h>

int main() {
    char ch, s[10], sen[20];

    scanf("%c\n", &ch);
    scanf("%s\n", &s);
    gets(sen);

    printf("%c\n%s\n%s", ch, s, sen);

    return 0;
}
```

pointers-in-c.c

```
void func(int *a, int *b) {
    int x = *a, y = *b;

    *a = x + y;
    *b = abs(x - y);
}

int main() {
    int a, b;
    scanf("%d %d", &a, &b);

    func(&a, &b);

    printf("%d\n", a);
    printf("%d", b);
    return 0;
}
```

printing-tokens.c

```
#include <stdio.h>

int main() {
    char str[1000];

    gets(str);

    for (int i = 0; i < strlen(str); i++) {
        if (str[i] == ' ') {
            printf("\n");
        } else {
            printf("%c", str[i]);
        }
    }

    return 0;
}
```

small-triangles-large-triangles.c

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

typedef struct triangle {
    int a;
    int b;
    int c;
} triangle;

int calcArea(int a, int b, int c, int n) {
    float area, p;

    p = (a + b + c) / 2.0;
    area = (p * (p - a) * (p - b) * (p - c));

    return area;
}

void sort_by_area(triangle *tr, int n) {
    for (int i = 0; i < n; i++) {
        for (int i = 0; i < n - 1; i++) {
            if (calcArea(tr[i].a, tr[i].b, tr[i].c, n) > calcArea(tr[i +
1].a, tr[i + 1].b, tr[i + 1].c, n)) {
                triangle aux;
                aux.a = tr[i].a;
                aux.b = tr[i].b;
                aux.c = tr[i].c;
                tr[i].a = tr[i + 1].a;
                tr[i].b = tr[i + 1].b;
                tr[i].c = tr[i + 1].c;
                tr[i + 1].a = aux.a;
                tr[i + 1].b = aux.b;
                tr[i + 1].c = aux.c;
            }
        }
    }
}

int main() {
    int n;
    scanf("%d", &n);
    triangle *tr = malloc(n * sizeof(triangle));
    for (int i = 0; i < n; i++) {
        scanf("%d%d%d", &tr[i].a, &tr[i].b, &tr[i].c);
    }
    sort_by_area(tr, n);
    for (int i = 0; i < n; i++) {
        printf("%d %d %d\n", tr[i].a, tr[i].b, tr[i].c);
    }
```

```
return 0;
}
```

sum-and-difference-of-two-numbers.c

```c
#include <stdio.h>

int main() {
    int ix, iy, isum, idff;
    float fx, fy, fsum,fdiff;

    scanf("%d %d %f %f", &ix, &iy, &fx, &fy);

    isum = ix + iy;
    idff = ix - iy;

    fsum = fx + fy;
    fdiff = fx - fy;

    printf("%d %d\n", isum, idff);
    printf("%.1f %.1f", fsum, fdiff);

    return 0;
}
```

sum-of-digits-of-a-five-digit-number.c

```
#include <stdio.h>

int main() {
    int n, nsum;

    scanf("%d", &n);

    nsum = (n / 10000) + (n % 10000 / 1000) + (n % 1000 / 100) + (n % 100
/ 10) + (n % 10 / 1);

    printf("%d", nsum);

    return 0;
}
```