

Отчёт №2

Лабораторные работы 5-8 вариант №15

5. лабораторная работа

```
#include<stdio.h>

int main() {

    int n = 9;
    int arr[9] = {99, 88, 77, 66, 55, 44, 33, 22, 11};

    for(int i = 0; i < n; i++){
        printf("arr[%d] = %d\n", i, arr[i]);
    }

    printf("-----");

    int r1 = 2, c1 = 2, r2 = 2, c2 = 2, i, j, p, m, q;

    int matr1[2][2] = {2, 5, 2, 2};
    int matr2[2][2] = {1, 2, 0, 1};
    int matr3[2][2] = {};

    printf("\nElements from the first matrix:");
    for(i = 0; i < r1; i++) {
        printf("\n");
        for(j = 0; j < c1; j++) {
            printf("%d ", matr1[i][j]);
        }
    }

    printf("\nElements from the second matrix:");
    for(i = 0; i < r2; i++) {
        printf("\n");
        for(j = 0; j < c2; j++) {
            printf("%d ", matr2[i][j]);
        }
    }

    int sum = 0;
    printf("\nThe product of our matrices is:\n");

    for(i = 0; i < 2; i++){
        for(j = 0; j < 2; j++){
            for(p = 0; p < 2; p++){
                sum = sum + matr1[i][p] * matr2[p][j];
            }
        }
    }
```

```

    matr3[i][j] = sum;
    sum = 0;
}
}

for(i = 0; i < 2; i++){
    for(j = 0; j < 2; j++){
        printf("%d ", matr3[i][j]);
    }
    printf("\n");
}

return 0;
}

```

6. лабораторная работа

```

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

int main(){

    const int n = 4;
    char *names[n];

    names[0] = 'W';
    names[1] = 'O';
    names[2] = 'R';
    names[3] = 'K';

    for(int i = 0; i < n; i++){
        printf("%c ", names[i]);
    }

    printf("\n-----\n");

    int n2 = 4;
    char *arrw;
    arrw = malloc(n2 * sizeof(int));

    arrw[0] = 'W';
    arrw[1] = 'O';
    arrw[2] = 'R';
    arrw[3] = 'K';

    for(int i = 0; i < n; i++){
        printf("%c ", arrw[i]);
    }

    free(arrw);

    return 0;
}

```

7. лабораторная работа

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

struct circle {
    float r;
    int no2;
};

int main(){

    enum music {classical, pop, rock, rap, newage, electronic};
    enum music x;
    x = rock;
    printf("The number of the rock music in our array is %d \n", x);

    struct triangle {
        int x1, y1, x2, y2, x3, y3;
    };

    struct triangle A = {1, 2, 3, 4, 5, 6};

    float AB = sqrt(pow((A.x2 - A.x1),2) + pow((A.y2 - A.y1),2));
    float BC = sqrt(pow((A.x3 - A.x2),2) + pow((A.y3 - A.y2),2));
    float AC = sqrt(pow((A.x3 - A.x1),2) + pow((A.y3 - A.y1),2));
    float P = AB + AC + BC;

    printf("\nThe perimeter of our triangle is %f", P);

    union status{
        struct{
            unsigned Active: 1;
            unsigned SDcard: 1;
            unsigned CompactFlashcard: 1;
            unsigned MemoryStickcard: 1;
        }bits;
        unsigned char data;
    };

    union status MyStatus;
    printf("\n\nEnter a hex number:");
    scanf("%x", &(MyStatus.data));
    printf("\nCard-reader status - ");
    printf("\nActive: %s", (MyStatus.bits.Active) ? "On" : "Off");
    printf("\nSD card: %s", (MyStatus.bits.SDcard) ? "On" : "Off");
    printf("\nCompact flash card: %s", (MyStatus.bits.CompactFlashcard) ?
"On" : "Off");
    printf("\nMemory stick card: %s", (MyStatus.bits.MemoryStickcard) ? "On"
: "Off");
    printf("\n\n");

    return 0;
```

```
• }

```

8. лабораторная работа

```
• #include<stdlib.h>
• #include<stdio.h>
• #include<string.h>
•
• int main() {
•
•     char first[100];
•     char second[100];
•     int n;
•     printf("Enter the first string -> ");
•     gets(first);
•     printf("Enter the second string -> ");
•     gets(second);
•     //2
•     printf("Input the number of first character -> ");
•     scanf("%d", &n);
•     char *c = strncat(first, second, n);
•     printf("The concatenation of our two strings is -> %s", c);
•     //5
•     char *c2 = strcpy(first, second);
•     printf("\nThe copy of one into the other is -> %s", c2);
•     //8
•     printf("\nEnter a character: ");
•     char sara;
•     scanf(" %c", &sara);
•     char *ss = strchr(first, sara);
•     printf("\nFinding the symbol in our string -> %p\n", ss);
•     //11
•     printf("Enter new strings: ");
•     char one[100];
•     char two[100];
•     printf("\n1\n");
•     scanf("\n %s", one);
•     printf("\n2\n");
•     scanf("\n %s", two);
•     /*char *s1 = strpbrk(one, two);
•     printf("\n The pointer to the first symbol which is similar to one of
the symbols 2nd string -> %p", s1);*/
•     int s2 = strspn(one, two);
•     printf("\nThe length of the initial segment of one which consists
entirely of characters in two -> %d, and the rest is -> ", s2);
•     /*int *s3 = strcmp(one, two);
•     printf("\nCompares the string pointed to, by str1 to the string pointed
to by str2. -> %d", s3);*/
•     //13
•     //char *s4 = strpbrk(one, two);
•     printf("\nEnter new strings: ");
•     char one1[100];
•     char two2[100];
•     printf("\n1 -> ");
•     scanf("\n %s", one1);
•     printf("\n2 -> ");

```

```
• scanf("\n %s", two2);
• char *s5 = strtok(one1, two2);
• //printf("something %s", *s5);
• int j = 100;
• char three[100];
• for(int i=0; i < 100; i++){
•     if(one1[i]!=*s5){
•         three[i] = one1[i];
•     } else{
•         three[i] = ' ';
•     }
• }
• printf("%s", one1);
•
•
•
• return 0;
• }
```