

Лабораторна робота №1

Виконав студент

Подоба Арсен

КН-108

1. Обчислити значення виразу при різних дійсних типах даних (float й double). Обчислення варто виконувати з використанням проміжних змінних. Порівняти й пояснити отримані результати.

Обчислити значення виразів. Пояснити отримані результати.

2. Програма розв'язку завдання №1:

а)

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

```
int main (void)
{
    float a,b,c,d,e,j;
```

```
    printf("Enter 'a': \n");
    scanf("%f", &a);
    printf("Enter 'b': \n");
    scanf("%f", &b);
```

```
    c = pow(a-b,4);
    d = (a*a*a*a-4*a*a*a*b);
    e = (6*a*a*b*b-4*a*b*b*b+b*b*b*b);
    j = (c-d)/e;
```

```
    printf("Your result %f\n", j);
```

```
    return 0;
```

б)

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

```
int main (void)
{
    double a,b,c,d,e,j;
```

```

printf("Enter 'a': \n");
scanf("%lf", &a);
printf("Enter 'b': \n");
scanf("%lf", &b);

c = pow(a-b,4);
d = (a*a*a*a-4*a*a*a*b);
e = (6*a*a*b*b-4*a*b*b*b+b*b*b*b);
j = (c-d)/e;

printf("Your result %lf\n", j);

return 0;
}

```

3. Результат роботи з даними типу float дорівнював 0
4. Результат роботи з даними типу double дорівнював 1
5. Результат роботи з даними типу double точніший, оскільки він дозволяє взяти більше знаків після коми. Оскільки в нас численні піднесення до степеня, а тип float заокруглює до певного значення після коми, результат отримується менш точнішим.
6. Програма розв'язку завдання №2:

а)

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

```

int main (void)
{
    int m,n,a;

    printf("Enter 'm': \n");
    scanf("%i", &m);
    printf("Enter 'n': \n");
    scanf("%i", &n);

    a = ++n * ++m;

    printf("Product of %i and %i is %i\n", m, n, a);

    return 0;
}

```

б)

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

```

int main (void)
{
    int m,n;
}

```

```

printf("Enter 'm': \n");
scanf("%i", &m);
printf("Enter 'n': \n");
scanf("%i", &n);

if (m++ < n)
    printf ("You entered correct numbers\n");
else
    printf ("You entered false numbers\n");

return 0;
}

```

В)

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

```

int main (void)
{
    int m,n;

    printf("Enter 'm': \n");
    scanf("%i", &m);
    printf("Enter 'n': \n");
    scanf("%i", &n);

    if (n++ > m)
        printf ("You entered right numbers\n");
    else
        printf ("You entered false numbers\n");

    return 0;
}

```

7. Результат роботи програми:

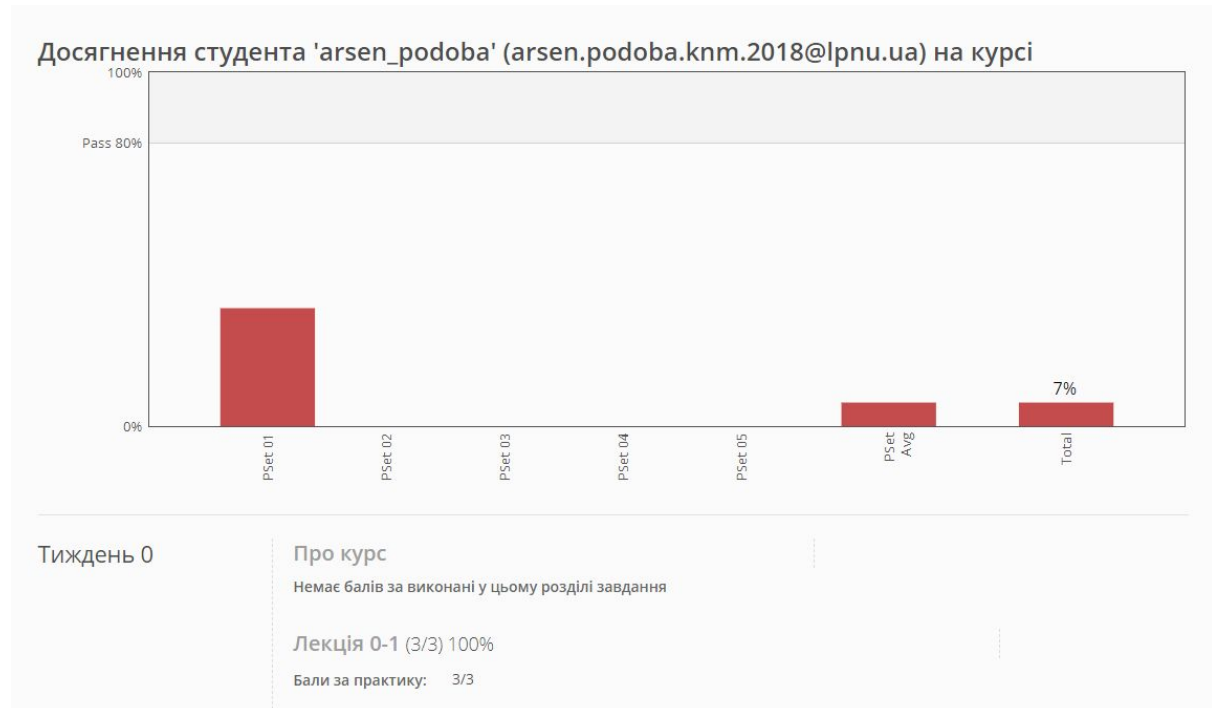
а) Програма додає до чисел введених користувачем 1 і множить їх між собою
 б) Програма додає до першого числа введеного користувачем 1 і порівнює його з другим. У випадку якщо перше менше вона видає позитивний результат. Навпаки негативний

в) Програма додає до другого числа введеного користувачем 1 і порівнює його з першим. У випадку якщо друге число більше вона видає позитивний результат. Навпаки негативний.

8. Пояснення результатів:

У цій програмі я використовув преінкримент і постінкримент, преінкримент спочатку збільшує число на 1 а потім порівнює, а постінкримент навпаки. Це справа з пріоритетності в мові програмуванні C.

Прогрес у CS50:



P.S. Сайт прометеус не приймає результати вже кілька днів тому прогрес не такий великий