Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту­­­­­



**Звіт**

про виконання

**Лабораторних та практичних робіт № 3**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.»

***Виконав:***

студент групи ШІ-11

Квіт Данило Романович

# **Тема роботи:**

Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.

# **Мета роботи:**

Покращити навички користування циклами та функціями. Додати до арсеналу поняття overload, рекурсію.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

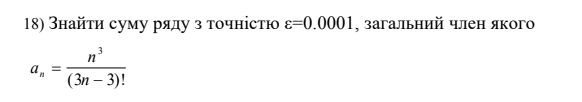
* Тема №1: Цикли.
* Тема №2: Функції.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли.
  + Джерела Інформації
    - [Стаття\_1](https://www.w3schools.com/cpp/cpp_for_loop.asp)
    - [Відео\_2:](https://www.youtube.com/watch?v=jjp3KdjIhxA)
    - [Відео\_3:](https://www.youtube.com/watch?v=vLnPwxZdW4Y)
    - [Стаття\_4](https://studfile.net/preview/5252553/page:3/)
  + Що опрацьовано:
    - Зрозумів використання вкладених циклів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30/11
  + Звершення опрацювання теми: 1/12
* Тема №2: Функції.
  + Джерела Інформації:
    - [Відео\_1](https://www.youtube.com/watch?v=-TkoO8Z07hI&t=17928s)
    - [Стаття\_2](https://www.w3schools.com/cpp/cpp_functions.asp)
    - [Стаття\_3](https://cplusplus.com/doc/tutorial/functions/)
  + Що опрацьовано:
    - Здобув використання: перевантаження функції, функція багатьох змінних
  + Статус: Ознайомлений
  + Початок опрацювання теми: 1/12
  + Звершення опрацювання теми: 1/12

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

VNS\_Lab\_2:

Варіант 18.

VNS\_Lab\_3:

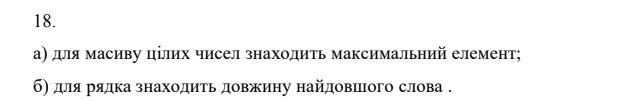
Варіант 18

A white paper with black text

Description automatically generated

VNS\_Lab\_7\_1:

Варіант 18



VNS\_Lab\_7\_2:

Варіант 18

A screenshot of a computer

Description automatically generatedClass\_Practice: Self\_Practice\_Task\_1:

A page of a document

Description automatically generated

Self\_Pratice\_Task\_2:

A white text with black text

Description automatically generated

## A diagram of a flowchart Description automatically generated

## **2. Дизайн та планована оцінка часу виконання завдань:**

Планована оцінка часу виконання: Flow\_Chart: Class\_Practice

VNS\_Lab\_2 : 20 хв

VNS\_Lab\_3 : 20 хв

VNS\_Lab\_7\_1 : 20 хв

VNS\_Lab\_7\_2 : 20 хв

Class\_Practice : 40 хв

Self\_Practice\_1: 10 хв

Self\_Practice\_2: 10 хв

## **3. Конфігурація середовища до виконання завдань:**

Середовище конфігуровано.

## **4. Код програм з посиланням на зовнішні ресурси:**

VNS\_Lab\_2:

#include <cstdio>  
using namespace std;  
double f(int x){  
 if(x==0) return 1.0;  
 return x\*f(x-1);  
}  
int main(){  
 int i = 1;  
 double ans = 0;  
 while(i\*i\*i/f(3\*i-3)>0.00001) {  
 ans += i \* i \* i / f(3 \* i - 3);  
 i++;  
 }  
 printf("%lf", ans);  
}

VNS\_Lab\_3:

#include <cmath>  
#include <cstdio>  
using namespace std;  
int main(){  
 float a = 0.1;  
 float b = 0.8;  
 float k = 10;  
 float step = (b-a)/k;  
 int n = 50;  
 double ans = 0;  
 for(float i=a;i<=b;i=i+step){  
 printf("%.2f primary: %.4f", i, 1.0/2 - 3.1415/4 \* abs(sin(i)));  
 for(int j = 1; j<=n; j++) {  
 ans += cos(2.0 \* j \* i)/(4\*j\*j - 1);  
  
 }  
 printf(" alternative: %0.4lf\n", ans);  
 ans = 0;  
 }  
 return 0;  
}

VNS\_Lab\_7\_1:

#include <iostream>  
using namespace std;  
int sum(int k, ...) {  
 int \*p = &k; //налаштували вказівник на параметр k  
 int s = 0;  
 p--;  
 for (; k != 0; k--)  
 s += \*(p+=2);  
 return s;  
}  
int nsd(int a, int b){  
 if(a==0||b==0) return max(a,b);  
 return nsd(min(a,b), max(a,b)%min(a,b));  
}  
int nsk2(int a, int b){  
 return a\*b/nsd(a,b);  
}  
int nsk(int k, ...){  
 int\*p = &k;  
 int ans = 1;  
 p+=3;  
 for(;k!=0;k--)  
 ans = nsk2(ans, \*(p+=2));  
  
 return ans;  
}  
int main(){  
 cout << nsk(3, 1, 2, 3) << '\n';  
 cout << nsk(5, 4, 2, 4, 2, 8) << '\n';  
 cout << nsk(6, 1, 2, 3, 4, 5, 6) << '\n';  
}

VNS\_Lab\_7\_2:

#include <iostream>  
#include <cstdio>  
using namespace std;  
int main() {  
 int n;  
 cout << "array len: ";  
 cin >> n;  
 int x;  
 cin >> x;  
 int ans = x;  
 for (int i = 1; i < n; i++){  
 cin >> x;  
 ans = max(ans,x);  
 }  
 cout << ans << '\n';  
   
   
 cout << "\nsentence:";  
 string s;  
 std::getline(cin, s);  
 int max\_word\_len = 0;  
 string max\_word;  
 int i=0;  
 int curr\_len=0;  
 string curr\_word = "";  
 while(i<s.size()){  
 if(s[i]==' '||i==s.size()-1){  
 if(curr\_len>max\_word\_len){  
 max\_word = curr\_word;  
 max\_word\_len = curr\_len;  
 }  
 curr\_len = 0;  
 curr\_word = "";  
 }else{  
 curr\_word+=s[i];  
 curr\_len++;  
 }  
 i++;  
 }  
 cout << max\_word;  
}

Class\_Practice:

#include <iostream>  
#include <string>  
#include <vector>  
using namespace std;  
void show(vector<string> b,vector<bool> a);  
void take(vector<string>& b, vector<bool>& a, string book);  
void ret(vector<string>& b, vector<bool>& a, string book);  
int main() {  
 bool running = true;  
 string command;  
 vector<string> books = **{**"book1","book2","book3"**}**;  
 vector<bool> availability**{**true, true, true**}**;  
 cout << "Manual:\n"  
 "exit\n"  
 "show\n"  
 "take <book>\n"  
 "return <book>\n";  
 while(running){  
 MainMenu:  
 cout << "Enter command: ";  
 cin >> command;  
 if(command=="show"){  
 show(books,availability);  
 do{cout << "Wanna continue? (y/n)\n";  
 cin >> command;  
 if(command=="n"){  
 running = false;  
 goto End;  
 }else if(command=="y") goto End;  
 }while(command!="n"||command!="y");  
 }else if(command=="exit"){  
 running=false;  
 continue;  
 }else if(command=="take"){  
 string name;  
 cout << "Book name: (0 to back)";  
 cin >> name;  
 if(name!="0")take(books,availability,name);  
 else goto MainMenu;  
 do{cout << "Wanna continue? (y/n)\n";  
 cin >> command;  
 if(command=="n"){  
 running = false;  
 goto End;  
 }else if(command=="y") goto MainMenu;  
 }while(command!="n"||command!="y");  
 }else if(command=="return"){  
 string name;  
 cout << "Book name: (0 to back)";  
 cin >> name;  
 if(name!="0")ret(books,availability,name);  
 else goto MainMenu;  
 do{cout << "Wanna continue? (y/n)\n";  
 cin >> command;  
 if(command=="n"){  
 running = false;  
 goto End;  
 }else if(command=="y") goto MainMenu;  
 }while(command!="n"||command!="y");  
 }  
 End:  
 }  
 return 0;  
}  
void show(vector<string> b,vector<bool> a){  
 int l = b.size();  
 for(string i:b) cout << i << '\n';  
}  
void take(vector<string>& b, vector<bool>& a, string book){  
 int l = b.size();  
 int b\_i = -1;  
 for(int i=0;i<l;i++) if(b[i]==book) b\_i = i;  
 if(b\_i==-1){  
 cout << "no such book\n";  
 }else{  
 if(a[b\_i]) a[b\_i]=false;  
 else cout << "book is not available\n";  
 }  
}void ret(vector<string>& b, vector<bool>& a, string book){  
 int l = b.size();  
 int b\_i = -1;  
 for(int i=0;i<l;i++) if(b[i]==book) b\_i = i;  
 if(b\_i==-1){  
 cout << "no such book\n";  
 }else{  
 if(!a[b\_i]) a[b\_i]=true;  
 else cout << "book is already in the library\n";  
 }  
}

Self\_Practice\_Task\_1

#include <iostream>  
using namespace std;  
  
long find\_nsd(long a, long b){  
 if(a%b==0)return b;  
 else if(b%a==0)return a;  
 else if(a>b) return find\_nsd(a%b,b);  
 else if(a<b) return find\_nsd(a,b%a);  
}  
int main(){  
 int n;  
 cin >> n;  
 long a[n];  
 for(int i=0;i<n;i++) cin >> a[i];  
 long answer=0;  
 long nsd=a[0];  
 for(int i=1;i<n;i++){  
 nsd=find\_nsd(nsd,a[i]);  
 }  
 for(int i=0;i<n;i++){  
 answer+=a[i]/nsd;  
 }  
 cout << answer;  
}

Self\_Practice\_Task\_2

#include <iostream>  
#include <cstdio>  
using namespace std;  
int main() {  
 int n;  
 cout << "array len: ";  
 cin >> n;  
 int x;  
 cin >> x;  
 int ans = x;  
 for (int i = 1; i < n; i++){  
 cin >> x;  
 ans = max(ans,x);  
 }  
 cout << ans << '\n';  
  
 cout << "\nsentence:";  
 char s[400];  
 int max\_word\_len = 0;  
 char max\_word[20];  
 int i=0;  
 int curr\_len=0;  
 char curr\_word[20];  
 fgets(s, 400, stdin);  
 int r=0;  
 while(s[i] != '\0'){  
 if(s[i]==' '||s[i+1]=='\0'){  
 if(curr\_len>max\_word\_len){  
 for(int j=0;j<20;j++) max\_word[j] = curr\_word[j];  
 max\_word\_len = curr\_len;  
 }  
 curr\_len = 0;  
 for(int j=0;j<20;j++) curr\_word[j] = '\0';  
 r = 0;  
 }else{  
 curr\_word[r]=s[i];  
 r++;  
 curr\_len++;  
 }  
 i++;  
 }  
 for(char j : max\_word){  
 if(j=='\0') break;  
 cout << j;  
 }  
}

## **5. Результати виконання завдань, тестування та фактично затрачений час:**

VNS\_Lab\_2:

A black screen with white text

Description automatically generated

Час затрачений на виконання завдання - 20хв

VNS\_Lab\_3:

A screen shot of a computer

Description automatically generated

Час затрачений на виконання завдання - 20хв

A number on a black background

Description automatically generatedVNS\_Lab\_7\_1

Час затрачений на виконання завдання - 40 хв

VNS\_Lab\_7\_2

array len: 3

51 3 25 dsc qwopkrs asd

51

sentence:qwopkrs

Час затрачений на виконання завдання - 30 хв

Class\_Practice\_

Manual:

exit

show

take <book>

return <book>

Enter command: show

book1

book2

book3

Wanna continue? (y/n)

y

Enter command: take

Book name: (0 to back)book1

Wanna continue? (y/n)

a

Wanna continue? (y/n)

s

Wanna continue? (y/n)

no

Wanna continue? (y/n)

y

Enter command: take book1

Book name: (0 to back)book is not available

Wanna continue? (y/n)

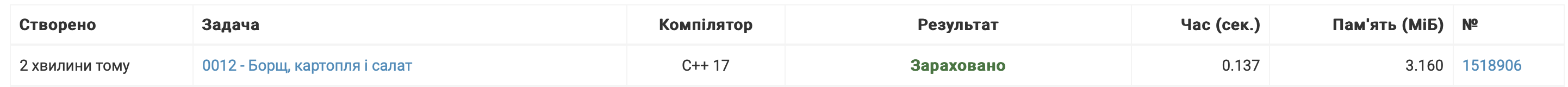
y

Enter command: exit

Process finished with exit code 0

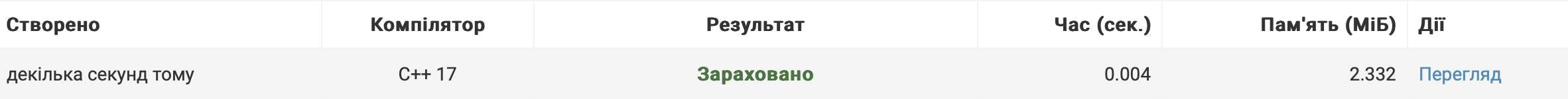
Час затрачений на виконання завдання - 30 хв

Self\_Practice\_1



Час затрачений на виконання завдання: 3хв

Self\_Practice\_2



Час затрачений на виконання завдання 15хв

## **6. Кооперація з командою:**

A group of people on a computer screen

Description automatically generated

# **Висновки:**

Навчився використовувати: цикли, функції, overloaded функції, вкладені цикли, рекурсію, функції багатьох змінних. Ознайомився з циклами for , for each, do while, while.