Lab 8

source code

ex 1:

ex 1.1:

```
#include <iostream>

int main()
{
    const int n = 5;
    int table[n+1];
    for(int i = 0; i < n+1; i++)
    {
        table[i]=i;
    }
    for(int i = 0; i < n+1; i++)
    {
        std::cout<< "index: " << i << " value: " << table[i] << std::endl;
    }
}
```

```
output:
        index: 0 value: 0
        index: 1 value: 1
        index: 2 value: 2
        index: 3 value: 3
        index: 4 value: 4
        index: 5 value: 5
```

ex 1.2:

```cpp
#include <iostream>
#include <vector>
#include <iomanip>

float function(float x)
{
    return x*x + 1;
}

int main()
{
    float step = 0.1;
    float range[2] = {-3.0 , 3.0};
    std::vector<float> funcargs;
    std::vector<float> funcvalues;

    for(float i = range[0]/step; i <= range[1]/step; i++)
    {
        funcargs.push_back(i*step);
        funcvalues.push_back(function(i*step));

    }

    for(int i = 0; i < funcargs.size(); i++)
    {
        std::cout<<std::setprecision(3)<<  "x: "<< funcargs[i] << " y: "<< funcvalues[i] << std:
    }

}
```

ex 1.3

```cpp
#include <iostream>

struct st
{
    int number;
    std::string name;
};

int main(){
    st tab[10];
    st *p = tab;
    for(int i = 0; i < 10; i++)
    {
        p = new st;
        p->name = "blank";
        p->number = i;
        tab[i] = *p;
        p++;
    }
    for(int i = 0; i < 10; i++)
    {
        std::cout<<tab[i].name<< " " << tab[i].number <<"\n";
    }
}
```

```
output:
blank 0
blank 1
blank 2
blank 3
blank 4
blank 5
blank 6
blank 7
blank 8
blank 9
```

ex 2

ex 2.1

```
#include <iostream>
using namespace std;

int main() {
        char tab[] = "Hello World!!!";
    char *p;

    for(int i = 0; i < sizeof(tab); i++)
    {
        if(tab[i] == 'W')
            {
                p = &tab[i];
                break;
            }
    }

    std::cout<<p<<"\n";

        cout << tab[0] << endl;                 // H
        cout << tab[1] << endl;                 // e
        cout << tab << endl;                    // Hello World!!!
        cout << sizeof(tab) << endl;    // 15

        char last = tab[sizeof(tab)-1];
        cout << int(last) << endl;              // 0
}
```

ex 2.2

```cpp
#include <iostream>

struct st
{
    int number;
    std::string name;
};

int main()
{
    st one, two;
    st *p = &one;

    p->name = "hello";
    p->number = 12;

    two.name = "world";
    two.number = 21;

    std::cout<< one.name << ' ' << one.number << "\n";
    std::cout<< two.name << ' ' << two.number << "\n";

}
```

output:

```
hello 12
world 21
```

ex 2.3

```cpp
#include <iostream>
using namespace std;

struct Product {
        int weight;
        float price;
};

int main() {
        Product p = {1, .5};
        Product *x = &p;

        p.weight = 2;
        x->weight = 4;

    Product tab[10];
    for(int i = 0; i < 10; i++)
    {
        tab[i] = *x;
    }
    Product *pTab;
    pTab = tab+4; // pTab = &tab[4];
    std::cout<<pTab->price<< ' ' << pTab->weight;

    // pTab points to 5th element of table tab
    // use it to print 5th element of the table

}
```

ex 2.4

```
#include <iostream>

struct pc
{
    std::string cpu;
    int ram;
};


int main()
{
    char *p_char;
    int *p_int;
    pc *p_pc;
    std::cout<< "size of pointer to char: "<< sizeof(p_char)*8 << " Bits\n";
    std::cout<< "size of pointer to int: "<< sizeof(p_int)*8 << " Bits\n";
    std::cout<< "size of pointer to structure: "<< sizeof(p_pc)*8 << " Bits\n";

}
```

```
output:
size of pointer to char: 64 Bits
size of pointer to int: 64 Bits
size of pointer to structure: 64 Bits
```

ex 3

ex 3.1

```
#include <iostream>
using namespace std;

int g_tab[] = {4, 3, 2, 1, 0};

int main(){
    int tab[] = {4, 3, 2, 1, 0};
    int *p = tab;
    int *g_p = g_tab;
    int size = sizeof(tab)/sizeof(int);
    std::cout<<"LOCAL SCOPE: \n";
    while(p < &tab[size])
    {
        std::cout<< p << ": " <<*p<<"\n";
        p++;
    }
    std::cout<<"GLOBAL SCOPE: \n";
    while(g_p < &g_tab[size])
    {
        std::cout<< g_p << ": " <<*g_p<<"\n";
        g_p++;
    }
}
```

```
output:
LOCAL SCOPE:
0xd555dffa50: 4
0xd555dffa54: 3
0xd555dffa58: 2
0xd555dffa5c: 1
0xd555dffa60: 0
GLOBAL SCOPE:
0x7ff73a814020: 4
0x7ff73a814024: 3
0x7ff73a814028: 2
0x7ff73a81402c: 1
0x7ff73a814030: 0
```

ex 3.2

```cpp
#include <iostream>

int main()
{
    char text[] = "This way to... adveture! (willump noises)";
    char *p = text;
    int length = (sizeof(text)-1)/sizeof(char);
    std::cout<< "length: " << length << " characters \n";
    std::cout<< p << "\n";
    while (p < &text[length])
    {
        std::cout<<*p<< ' ';
        p++;
    }

}
```

```
output:
length: 41 characters
This way to... adveture! (willump noises)
T h i s   w a y   t o . . .   a d v e t u r e !   ( w i l l u m p   n o i s e s )
```

ex 3.3

```cpp
#include <iostream>
using namespace std;

int main(){
    char tab[] = "Ala ma kota";
    char *p = tab;

    float array[] = {3.14, 1.41, 2.71};
    float *p_a = array;

    while ( *p ) {
        cout << *(p++);   // *(p++)
    }
    cout << endl;

    while ( p_a < &array[sizeof(array)/sizeof(float)])
    {
        std::cout<<*p_a++ << ' ';
    }
}

// result: Ala ma kota
// while( p ) {}  what kind of mistake?
```

output:

```
Ala ma kota
3.14 1.41 2.71
```

ex 3.4

```cpp
#include <iostream>
using namespace std;

void print_tab(int *tab, int size){
    for(int i = 0; i < size; i++)
    {
        std::cout<<*tab++<<", ";
    }
    std::cout<<"\n";
}

int main(){
    int tab[11];
    int *start = tab;
    int *end = &tab[10];
    int *init = tab;
    int size = sizeof(tab)/sizeof(int);
    int x = 0;
    while (init <= end)
    {
        *init = x;
        init++;
        x++;
    }
    std::cout<<"NOT REVERSED: \n";
    print_tab(tab, size);

    while (end > start) {
        int tmp = *end;
        *end = *start;
        *start = tmp;
        end--;
        start++;
    }
    std::cout<<"REVERSED: \n";
    print_tab(tab, size);
}
```

```
output:
NOT REVERSED:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
REVERSED:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
```

ex 3.5

```
#include <iostream>
using namespace std;

int main(){
    int x = 10;
    int *p;

    if (x > 5) {
        p = new int;
        *p = x*10;
    }

    cout << *p << endl;

    delete p;
}
```

I can't find any error to correct in this code, only thing that could couse en error is pointer `p` which is uninitialization is case of `x < 5`