Cover:
Student name: Chea Ilong
Group: 1
ID: 100022
Lab1 and 2
Advance Algorithm

Lab1-ex1: Find root of quadratic equation ax2+bx+c=0. You need to study the case delta is 0, delta is greater than 0, and delta is less than 0.

```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    double a, b, c, delta, X1, X2;

    cout << "Enter coefficients a, b, and c: ";
    cin >> a >> b >> c;
    cout << a << "x^2" << " + " << b << "x" << " + " << c << " = 0" << endl;
    delta = b * b - 4 * a * c;

    if (delta > 0)
    {
        X1 = (-b + sqrt(delta)) / (2 * a);
        X2 = (-b - sqrt(delta)) / (2 * a);
        cout << "Roots are real and different." << endl;
        cout << "X1 = " << X1 << endl;
        cout << "X2 = " << X2 << endl;
    }
    else if (delta == 0)
    {
        X1 = X2 = -b / (2 * a);
        cout << "Roots are real and the same." << endl;
        cout << "X1 = X2 = " << X1 << endl;
    }
    else
    {

        cout << "the equation has no real roots" << endl;
    }

    return 0;
}

```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\MSI PC\Desktop\T1Y2> cd "c:\Users\MSI PC\Desktop\T1Y2\" ; if ($?) { g++ Exercise1.cpp -o Exercise1 } ; if ($?) { .\Exercise1 }
Enter coefficients a, b, and c: 1 -5 6
1x^2 + -5x + 6 = 0
Roots are real and different.
X1 = 3
X2 = 2
PS C:\Users\MSI PC\Desktop\T1Y2>
```

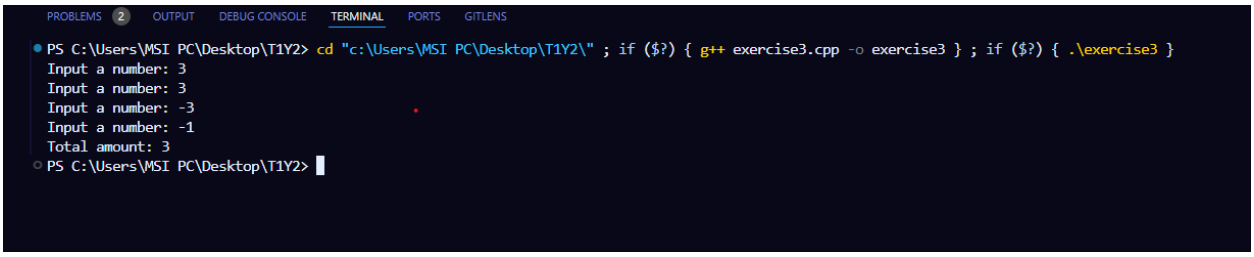
Lab1-ex2: Display numbers 1 to 1000 on the screen except the numbers 100, 200, 300, 400 and 500.

```cpp
#include <iostream>
using namespace std;
int main()
{

    for (int i = 1; i <= 1000; i++)
    {
        if (i == 100 || i == 200 || i == 300 || i == 400 || i == 500)
        {
            continue;
        }
        cout << i << " ";
    }

    return 0;
}
```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163
164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254
255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345
346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390
391 392 393 394 395 396 397 398 399 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481
482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527
528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572
573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617
618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662
663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707
708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752
753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797
798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842
843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887
888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932
933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977
978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000
PS C:\Users\MSI PC\Desktop\T1Y2\output>

C/C++ ...
C/C++ Com...
```

Lab1-ex3: Ask a user to input a number. Keep asking the user for more numbers until the user inputs -1. Display the total summation of all input numbers except -1.

```cpp
#include <iostream>
using namespace std;
int main()
{
    int n = 0, sum = 0;

     while (n != -1)
     {
         cout << "Input a number: ";
         cin >> n;

         if (n != -1)
         {
             sum += n;
         }
     }

     cout << "Total amount: " << sum;

     return 0;
}
```



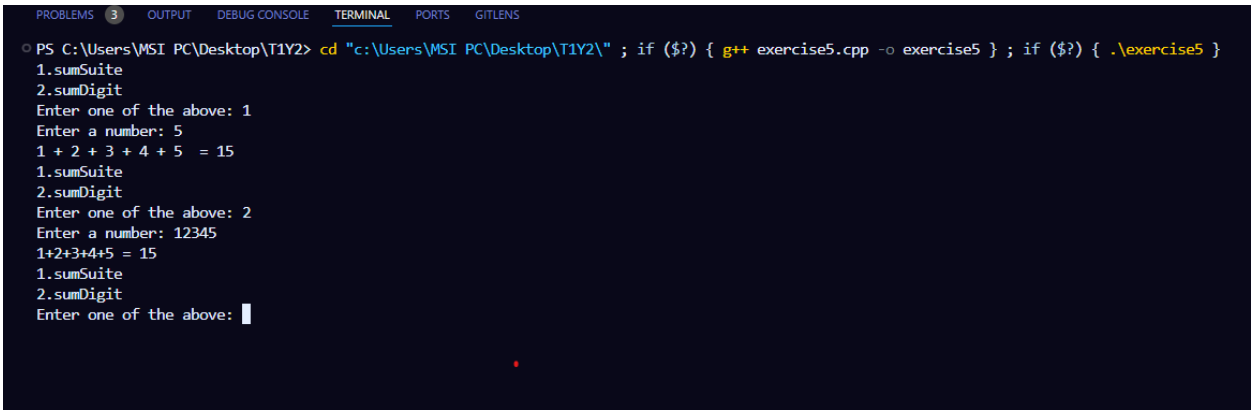
Lab1-ex4: Write a function to display and compute this suit 1/1 + 1/2 + … + 1/n , where n is the parameter of this function.

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

void exercise4(int n)
{
    float sum = 0;
    for (int i = 1; i <= n; i++)
    {
        sum += 1.0 / i;
        cout << "1/" << i << " + ";
    }
    cout << "\b\b= " << sum << endl;
}

int main()
{

    int n;

    cout << "Enter the value of n: ";
    cin >> n;

    exercise4(n);

    return 0;
}

```

Lab1-ex5: a: a summation function to calculate the sum of the first n integer 1+2+3+…+n

B: sum digits of a number

```cpp
#include <iostream>
#include <algorithm>
using namespace std;
int sumSuite(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
    {
        cout << i << " ";
        if (i != n)
        {
            cout << "+ ";
        }
            sum += i;
    }
    cout  << " = " << sum << endl;
    return sum;
}

int sumDigit(int n)
{
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        sum += n % 10;
        n /= 10;
    }

    return sum + n;
}

void printSum(int n)
{
    int count = 0;
    int num = n;
    int a[100];
    while (n != 0)
    {
        a[count] = n % 10;
        n = n / 10;
        count++;
    }

    int count_v2 = count - 1;
    while (count > 0)
    {
        cout << a[count_v2];
        if (count > 1)
        {
            cout << "+";
        }
        count--;
        count_v2--;
    }
}
int main()
{
    int userInput;
    int one;
    while (1)
    {
        cout << "1.sumSuite" << endl;
        cout << "2.sumDigit" << endl;
        cout << "Enter one of the above: ";
        cin >> one;
        if (one > 2 || one < 0){
            cout << "Please enter the choice above!!!" << endl;
        }
        switch (one)
        {
        case 1:
            cout << "Enter a number: ";
            cin >> userInput;
            sumSuite(userInput);
            break;
        case 2:
            cout << "Enter a number: ";
            cin >> userInput;
            printSum(userInput);
            cout << " = "<< sumDigit(userInput) << endl;
            break;
        }
    }

    return 0;
}

```



Lab1-ex6:create a struct of people. That contain information such as name, age, zodiac. And an array that can store person information up to 20 then:

1. Ask the user to input the information for 4 people and store in an array
2. Display the information for each person on screen
3. Show the information of the person with the oldest age

```cpp
#include <iostream>
using namespace std;
struct person
{
    string name;
    int age;
    string zodiac_sign;
};

int main()
{

    int n = 0;
    do
    {
        if (n > 20)
        {
            cout << "The max number is 20!!\n";
        }
        cout << "How many people to store (max:20): ";
        cin >> n;
    } while (n > 20);
    person people[n];

    for (int i = 0; i < n; i++)
    {
        cout << "Person " << i + 1 << endl;
        cout << "Name: ";
        cin.ignore();
        getline(cin, people[i].name);
        cout << "Age: ";
        cin >> people[i].age;
        cout << "Zodiac sign: ";
        cin >> people[i].zodiac_sign;
    }

    int oldest = 0;
    for (int i = 0; i < n; i++)
    {
        if (people[i].age > oldest)
        {
            oldest = people[i].age;
        }
    }

    for (int i = 0; i < n; i++)
    {
        cout << "======================\n";
        cout << "Person" << i + 1 << endl;
        cout << "Name: " << people[i].name << endl;
        cout << "Age: " << people[i].age << endl;
        cout << "Zodiac sign: " << people[i].zodiac_sign << endl;
    }
    for (int i = 0; i < n; i++)
    {
        if (people[i].age == oldest)
        {
            cout << "======================\n";
            cout << "Oldest person" << endl;
            cout << "Name: " << people[i].name << endl;
            cout << "Age: " << people[i].age << endl;
            cout << "Zodiac sign: " << people[i].zodiac_sign << endl;
        }
    }
    return 0;
}
```

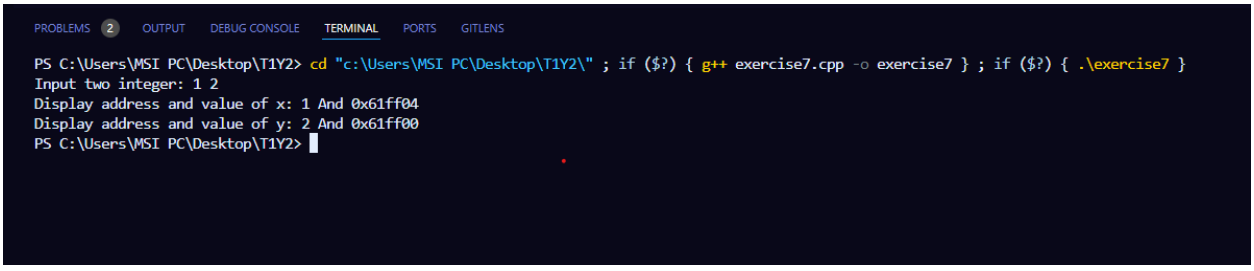
```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\MSI PC\Desktop\T1Y2> cd 'c:\Users\MSI PC\Desktop\T1Y2\output'
PS C:\Users\MSI PC\Desktop\T1Y2\output> & .\'exercise6.exe'
How many people to store (max:20): 21
The max number is 20!!
How many people to store (max:20): 4
Person 1
Name: chea ilong
Age: 18
Zodiac sign: leo
Person 2
Name: rith
Age: 19
Zodiac sign: pices
Person 3
Name: nuth
Age: 20
Zodiac sign: gemini
Person 4
Name: joker
Age: 21
Zodiac sign: aries
======================
Person1
Name: chea ilong
Age: 18
Zodiac sign: leo
======================
Person2
Name: rith
Age: 19
Zodiac sign: pices
======================
Person3
Name: nuth
Age: 20
Zodiac sign: gemini
======================
Person4
Name: joker
Age: 21
Zodiac sign: aries
======================
Oldest person
Name: joker
Age: 21
Zodiac sign: aries
PS C:\Users\MSI PC\Desktop\T1Y2\output>
```

Lab1-ex7:

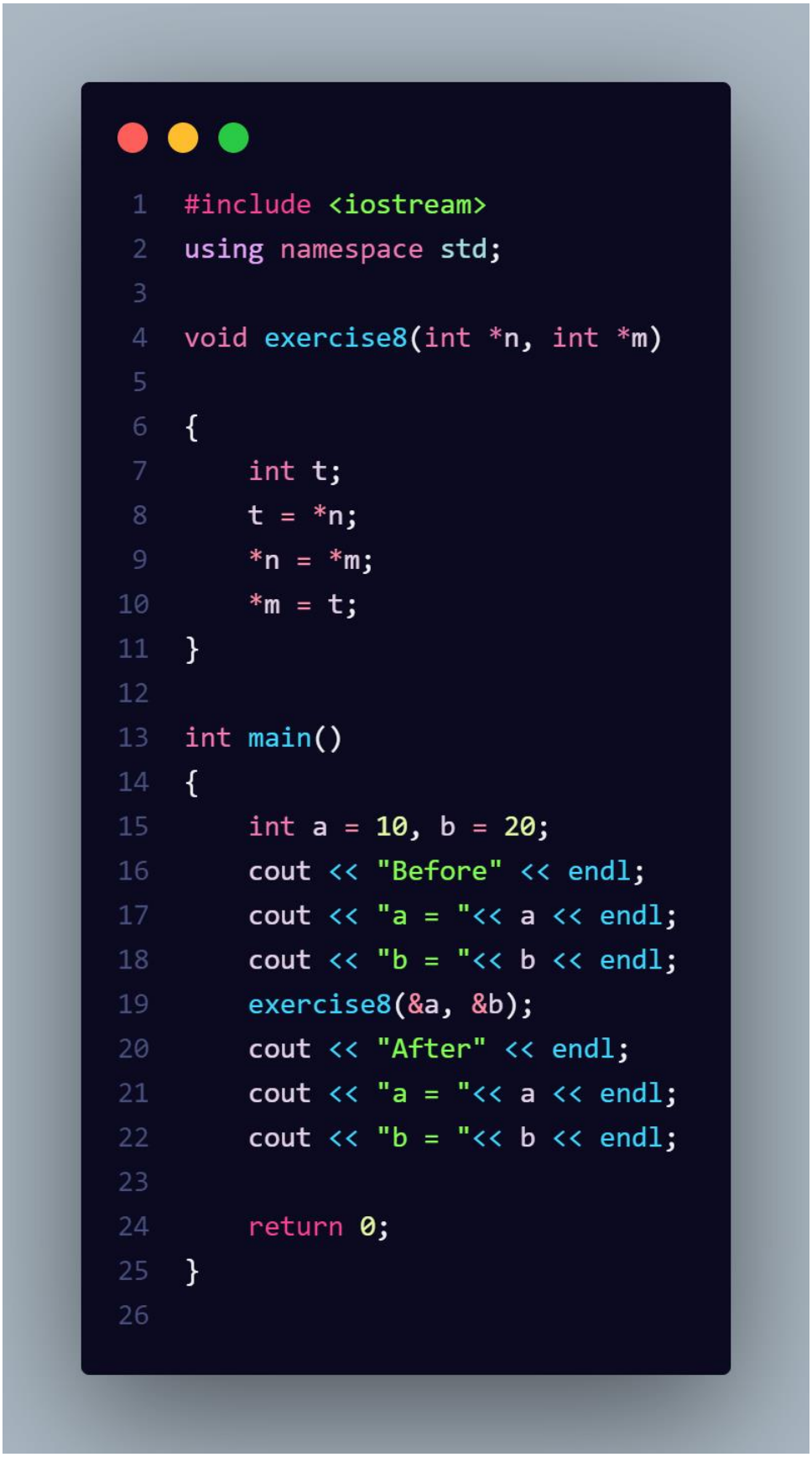
Get two integer x and y.

1. Display address and value of x using p1
2. Display address and value of y using p2

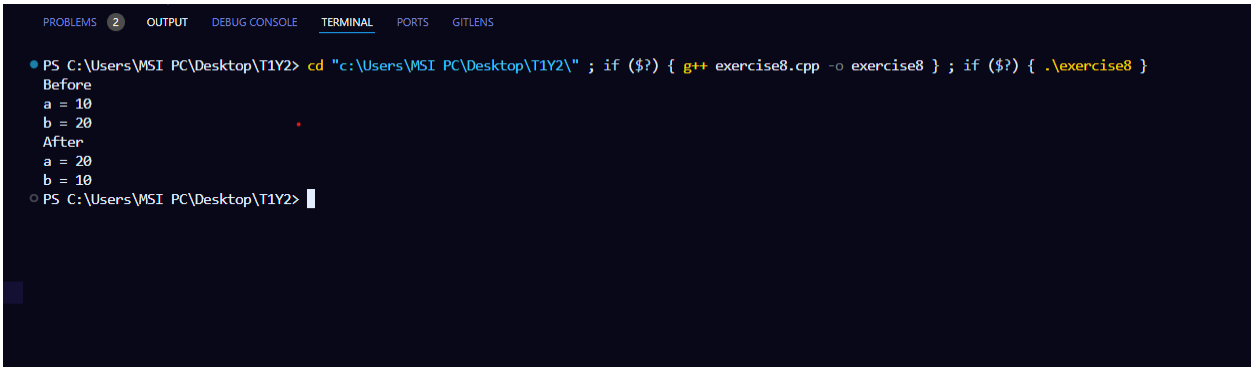
```cpp
#include <iostream>
using namespace std;
int main()
{

    int x, y;

    cout << "Input two integer: ";
    cin >> x >> y;

    int *p1 = &x;
    int *p2 = &y;

    cout << "Display address and value of x: " << *p1 << " And " << &p1 << endl;
    cout << "Display address and value of y: " << *p2 << " And " << &p2 << endl;

    return 0;
}
```



Lab1-ex8: create a function that can exchange two number.

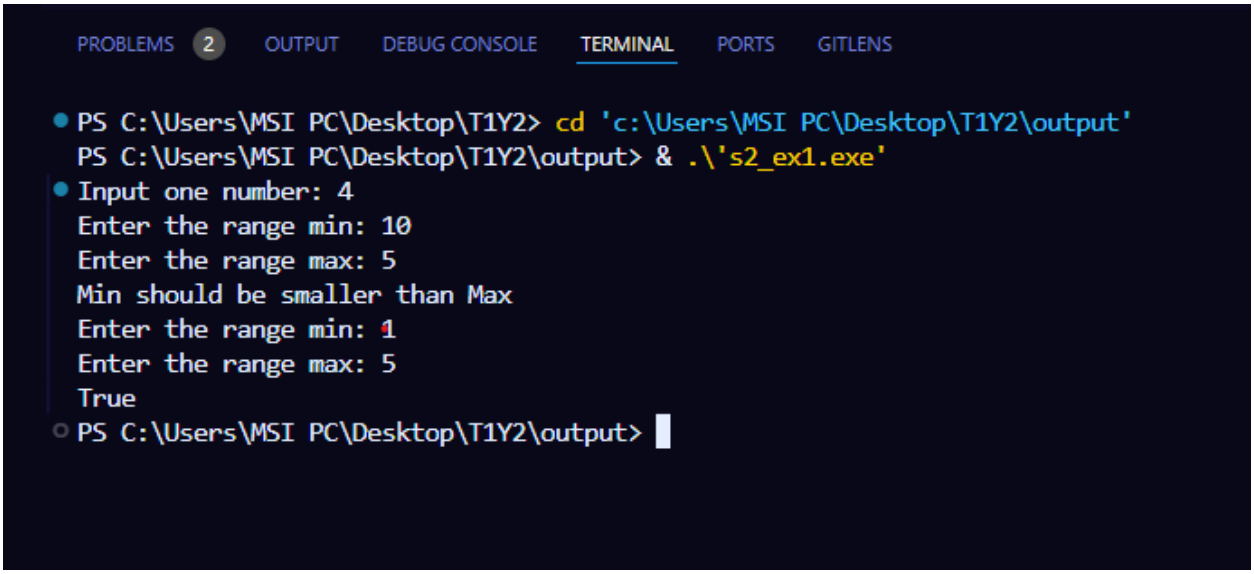


```cpp
#include <iostream>
using namespace std;

void exercise8(int *n, int *m)

{
    int t;
    t = *n;
    *n = *m;
    *m = t;
}

int main()
{
    int a = 10, b = 20;
    cout << "Before" << endl;
    cout << "a = "<< a << endl;
    cout << "b = "<< b << endl;
    exercise8(&a, &b);
    cout << "After" << endl;
    cout << "a = "<< a << endl;
    cout << "b = "<< b << endl;

    return 0;
}

```



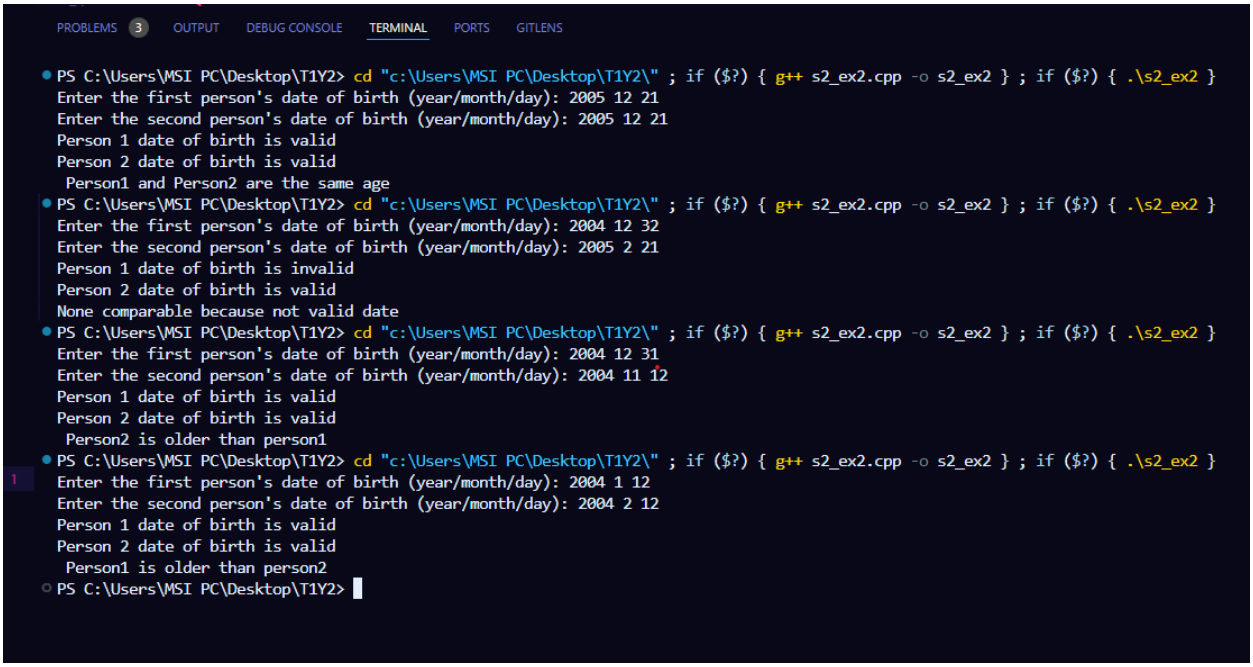
Lab2-ex1: We want to check if a given number is inside a given range. 1- Input the number and the range defined by a minimum and maximum 2- Output whether the number is "inside" or "outside" the range

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

/**
 * Return whether the given number if inside the given range
 * @param number : the number
 * @param min : the range min
 * @param max : the range max
 * @return true if inside the range, false otherwise
 */
bool isInside(int number, int min, int max)
{
    // TODO

    if (number >= min && number <= max)
    {
        return true;
    }
}

int main()
{

    // 1-  Input the number
    // TODO
    while (1){
    int min, max, number;

    cout << "Input one number: ";
    cin >> number;

    // 2 - Input the range
    do
    {
        if (min > max)
        {
            cout << "Min should be smaller than Max" << endl;
        }
        cout << "Enter the range min: ";
        cin >> min;

        cout << "Enter the range max: ";
        cin >> max;

    } while (min > max);
    // BONUS : If the user accidentally enters a minimum value that is larger
    // than the maximum value, ask them to enter the values again.

    // TODO

    // 3-  Check if the number is inside the range
    if (isInside(number, min, max) == true)
    {
        cout << "True\n";
    }
    else
    {
        cout << "False\n";
    }
    // TODO
    }
    return 0;
}
```



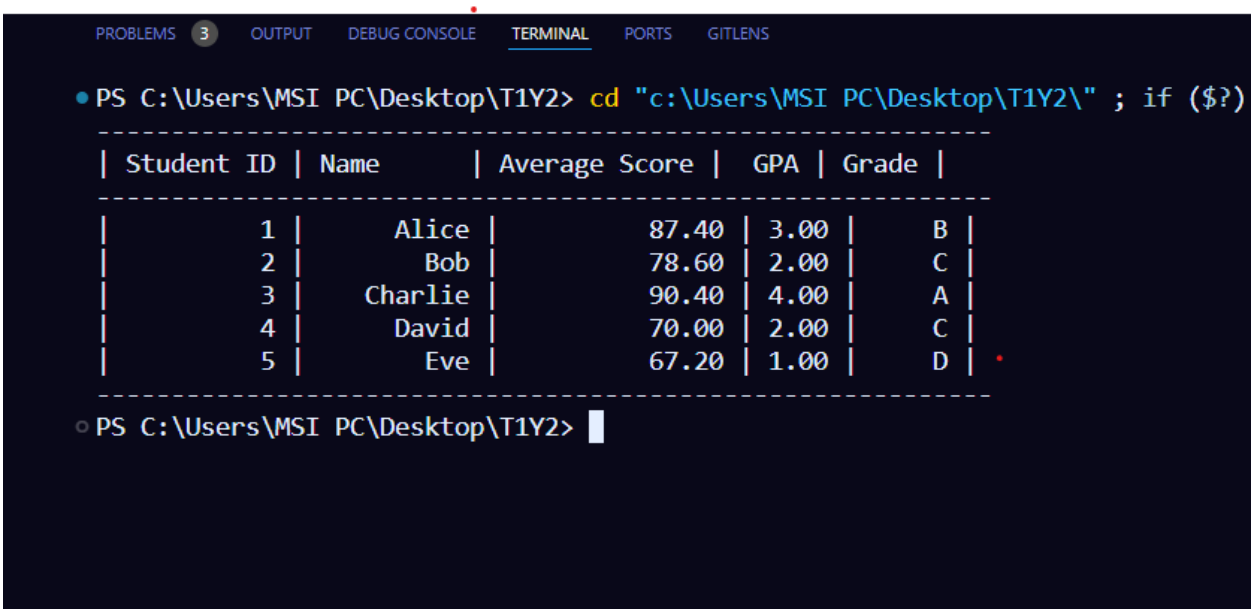
Lab2-ex2: We want to compare the ages of two people based on their dates of birth and determines which person is the youngest. 1- Input the date of birth 2- Output which person is the youngest or if they are of the same age.

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

// Function to check if a year is a leap year
bool isLeapYear(int year)
{
    return (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
}

/**
 * Validate a given date
 * @param year, month, day : the date
 * @return true if the date is valid, false otherwise
 */
bool isValidDate(int year, int month, int day)
{
    // TODO
    // The year shall be a positive number
    // The month shall be a number between 1 and 12
    // The day shall be a number between 1 and the max days
    int max;
    if (month == 2 && isLeapYear(year))
    {
        max = 29;
    }
    else if (month == 2)
    {
        max = 28;
    }
    else if (month == 4 || month == 6 || month == 9 || month == 11)
    {
        max = 30;
    }
    else
    {
        max = 31;
    }

    return (year > 0 && month >= 1 && month <= 12 && day >= 1 && day <= max);
}

/**
 * Compare 2 dates (date1 and date2)
 * @param year1, month1, day1 : date 1
 * @param year2, month2, day2 : date 2
 * @return
 *      -1 if date1 < date2
 *      0 if they are the same
 *      1 if date1 > date2
 */
int compare(int year1, int month1, int day1, int year2, int month2, int day2)
{
    // TODO
    if (year1 > year2)
    {
        return 1;
    }
    else if (year1 < year2)
    {
        return -1;
    }
    else
    {

        if (month1 > month2)
        {
            return 1;
        }
        else if (month1 < month2)
        {
            return -1;
        }
        else
        {
            if (day1 > day2)
            {
                return 1;
            }
            else if (day1 < day2)
            {
                return -1;
            }
            else if (day1 == day2)
            {
                return 0;
            }
        }
    }
}

int main()
{
    // 1- Input the dates
    int year1, month1, day1;
    cout << "Enter the first person's date of birth (year/month/day): ";
    cin >> year1 >> month1 >> day1;

    int year2, month2, day2;
    cout << "Enter the second person's date of birth (year/month/day): ";
    cin >> year2 >> month2 >> day2;

    // 2- Validate the dates (BONUS)
    // TODO
    if (isValidDate(year1, month1, day1))
    {
        cout << "Person 1 date of birth is valid" << endl;
    }
    else
    {
        cout << "Person 1 date of birth is invalid" << endl;
    }
    if (isValidDate(year2, month2, day2))
    {
        cout << "Person 2 date of birth is valid" << endl;
    }
    else
    {
        cout << "Person 2 date of birth is invalid" << endl;
    }

    // 3- Compare the dates
    // TODO
    if ((isValidDate(year1, month1, day1)) && (isValidDate(year2, month2, day2)))
    {
        if (compare(year1, month1, day1, year2, month2, day2) == 1)
        {
            cout << " Person1 is older than person2";
        }
        else if (compare(year1, month1, day1, year2, month2, day2) == -1)
        {
            cout << " Person2 is older than person1";
        }
        else
        {
            cout << " Person1 and Person2 are the same age";
        }
    }
    else
    {
        cout << "None comparable because not valid date";
    }
    return 0;
}

```



Lab2-ex3: We want, for each student of our database: • To calculate their average score • To calculate their GPA • To calculate their grade letters

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

#define DECIMAL_DIGIT 2
struct Student
{
    int id;
    std::string name;
    std::vector<int> scores;
};

std::vector<Student> students = {
    {1, "Alice", {85, 92, 78, 94, 88}},
    {2, "Bob", {70, 81, 73, 85, 79}},
    {3, "Charlie", {93, 89, 84, 91, 95}},
    {4, "David", {65, 70, 75, 60, 80}},
    {5, "Eve", {54, 68, 72, 77, 65}}};

/**
 * Calculate the average score of the given student
 * @param Student : the student
 * @return the average of students scores
 */
float calculateAverage(Student student)
{
    // TODO
    float sum = 0.0;
    int size = student.scores.size();
    for (int i = 0; i < size; i++)
    {
        sum += student.scores[i];
    }
    return sum / size;
}

/**
 * Calculate GPA based on average score
 * @param Student : the student
 * @return the average of students scores
 */
float calculateGPA(Student student)
{

    // TODO
    float check = calculateAverage(student);
    float gpa = 0;

    if (check >= 90 && check <= 100)
    {
        gpa = 4.0;
    }
    else if (check >= 80 && check <= 89)
    {
        gpa = 3.0;
    }
    else if (check >= 70 && check <= 79)
    {
        gpa = 2.0;
    }
    else if (check >= 60 && check <= 69)
    {
        gpa = 1.0;
    }
    else
    {
        gpa = 0.0;
    }
    return gpa;
}

/**
 * Calculate letter grade based on GPA
 * @param Student : the student
 * @return the average of students scores
 */
std::string calculateGrade(Student student)
{
    // TODO
    std::string grade;
    float check2 = calculateGPA(student);

    if (check2 == 4.0)
    {
        grade = "A";
    }
    else if (check2 == 3.0)
    {
        grade = "B";
    }
    else if (check2 == 2.0)
    {
        grade = "C";
    }
    else if (check2 == 1.0)
    {
        grade = "D";
    }
    else
    {
        grade = "E";
    }
    return grade;
}

/**
 * Compute and print students statistic (average, GPA, grade letter) on console
 * @param Student : the student
 */
void printStudentStatistics(const Student &student)
{
    float average = calculateAverage(student);
    float gpa = calculateGPA(student);
    std::string grade = calculateGrade(student);

    std::cout << "| " << std::setw(10) << student.id << " | "
              << std::setw(10) << student.name << " | "
              << std::setw(14) << average << " | "
              << std::setw(4) << gpa << " | "
              << std::setw(5) << grade << " |\n";
}

int main()
{

    std::cout << std::fixed << std::setprecision(DECIMAL_DIGIT); // Set decimal precision for average and GPA
    std::cout << "                                                                \n";
    std::cout << "| Student ID | Name       | Average Score |  GPA | Grade |\n";
    std::cout << "-----------------------------------------------------------\n";

    // Call the function printStudentStatistics for each student
    // TODO
    int Size = students.size();

    for (int i = 0; i < Size; i++)
    {
        printStudentStatistics(students[i]);
    }
    std::cout << "-----------------------------------------------------------\n";

    return 0;
}
```


PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
PS C:\Users\MSI PC\Desktop\T1Y2> cd "c:\Users\MSI PC\Desktop\T1Y2\" ; if ($?)
Student ID	Name	Average Score	GPA	Grade
1	Alice	87.40	3.00	B
2	Bob	78.60	2.00	C
3	Charlie	90.40	4.00	A
4	David	70.00	2.00	C
5	Eve	67.20	1.00	D
PS C:\Users\MSI PC\Desktop\T1Y2>