Opis programu

Program jest grą, która polega na wpisywaniu przez użytkownika indeksów losowo generowanych liczb (bez powtórzeń) do tablicy o określonej długości (poziomie trudności). Gracz musi podawać indeksy liczb, w taki sposób, aby wszystkie liczby w tablicy znajdowały się w kolejności rosnącej. Następnie program sprawdza, czy użytkownik wpisał prawidłowy indeks oraz czy została zachowana kolejność rosnąca wszystkich liczb.

Zasada działania programu

1. Inicjalizacja gry.

Program daje możliwość użytkownikowi na wybranie poziomu trudności gry, poprzez podanie liczby miejsc tablicy. Pozwala na to konstruktor parametrowy RandomGame(int difficulty) (zdjęcie 1a, zdjęcie 1b).

```
RandomGame(int difficulty) {
    this->difficulty = difficulty;
    tab = new int[difficulty];
    Reset();
}
```

zdjęcie 1a

```
int main()
{
    RandomGame game(5);
}
```

zdjęcie 1b

Jeśli użytkownik nie określił liczby miejsc, program domyślnie ustawia długość tablicy na 5 elementów. Pozwala na to konstruktor domyślny RandomGame() (zdjęcie 2a, zdjęcie 2b).

```
RandomGame() {
    difficulty = 5;
    tab = new int[difficulty];
    Reset();
}
```

zdjęcie 2a

```
int main()
{
    RandomGame game;
}
```

zdjęcie 2b

Należy również pamiętać, aby zwolnić pamięć zaalokowaną dla tablicy. Służy do tego destruktor ~RandomGame() (zdjęcie 3).

```
~RandomGame() {
    delete[] tab;
}
```

zdjęcie 3

Następnie program tworzy dynamiczną tablicę i resetuje jej zawartość, ustawiając we wszystkich miejscach wartość -1. Pozwala na to funkcja Reset(), która jest wykonywana przy każdej inicjalizacji gry (zdjęcie 4).

```
void Reset() {
    for(int i = 0; i < difficulty; i++) {
        tab[i] = -1;
    }
}
```

zdjęcie 4

2. Rozpoczęcie gry.

Za całą logikę gry odpowiedzialna jest funkcja Play(). Program wchodzi w pętlę, losuje liczbę z zakresu od 0 do 100 i sprawdza, czy nie została ona już wcześniej wylosowana, wykorzystując funkcję isInUsedNumbers(int number), która zwraca logiczny typ danych (zdjęcie 5).

```
bool isInUsedNumbers(int number) {
    for(int x : usedNumbers) {
        if(x == number) {
            return true;
        }
    }
    return false;
}
```

zdjęcie 5

Jeśli liczba nie została wcześniej wylosowana, program dodaje ją do wektora usedNumbers, aby uniknąć powtórzeń w kolejnych losowaniach (zdjęcie 6).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }
        usedNumbers.push_back(random_number);
    }
}
```

zdjęcie 6

Następnie program za pomocą funkcji Display() wyświetla aktualny stan tablicy, pokazując które miejsca są już zajęte, a które puste (zdjęcie 7).

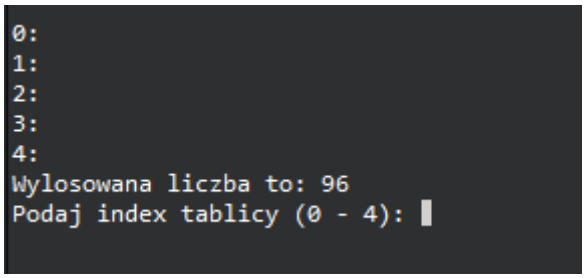
```
void Display() {
    for(int i = 0; i < difficulty; i++) {
        cout << i << ": ";
        if(tab[i] == -1) {
            cout << "" << endl;
        }
        else {
            cout << tab[i] << endl;
        }
    }
}
```

zdjęcie 7

Pod wyświetloną tablicą program wypisuje aktualnie wylosowaną liczbę oraz pyta użytkownika pod którym indeksem ma umieścić daną liczbę (zdjęcie 8a, zdjęcie 8b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }
        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;
    }
}
```

zdjęcie 8a



zdjęcie 8b

Po podaniu indeksu przez gracza program przechodzi do sprawdzenia poprawności wpisanej liczby. Program sprawdza czy:

- podany indeks jest mniejszy od 0 oraz większy od liczby miejsc w tablicy. Jeśli tak, oznacza to, że gracz podał niepoprawny indeks i gra od razu się kończy, wyświetlając komunikat: „Podano niepoprawny index. Koniec gry” (zdjęcie 9a, zdjęcie 9b).

- podany indeks jest już zajęty. Jeśli jest, oznacza to, że gracz podał niepoprawny indeks i gra od razu się kończy, wyświetlając komunikat: „Podano niepoprawny index. Koniec gry” (zdjęcie 9a, zdjęcie 9b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }
        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;
        if (index < 0 || index >= difficulty || tab[index] != -1) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }
    }
}
```

zdjęcie 9a

```
0:
1:
2:
3:
4:
Wylosowana liczba to: 96
Podaj index tablicy (0 - 4): 5

Process exited with code: 0.Podano niepoprawny index. Koniec gry
```

zdjęcie 9b

- liczba w tablicy przed podanym indeksem jest większa niż obecnie wylosowana liczba. Jeśli tak, oznacza to, że liczba pod indeksem podanym przez gracza nie zachowuje kolejności rosnącej i gra od razu się kończy, wyświetlając komunikat: „Podano niepoprawny index. Koniec gry” (zdjęcie 10a, zdjęcie 10b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }

        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;

        if (index < 0 || index >= difficulty || tab[index] != -1) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index > 0 && tab[index - 1] != -1 && tab[index - 1] > random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }
    }
}
```

zdjęcie 10a

```
0:
1:
2: 69
3:
4:
Wylosowana liczba to: 37
Podaj index tablicy (0 - 4): 3

Process exited with code: 0.Podano niepoprawny index. Koniec gry
```

zdjęcie 10b

- liczba w tablicy za podanym indeksem jest mniejsza niż obecnie wylosowana liczba. Jeśli tak, oznacza to, że liczba pod indeksem podanym przez gracza nie zachowuje kolejności rosnącej i gra od razu się kończy, wyświetlając komunikat: „Podano niepoprawny index. Koniec gry” (zdjęcie 11a, zdjęcie 11b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }

        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;

        if (index < 0 || index >= difficulty || tab[index] != -1) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index > 0 && tab[index - 1] != -1 && tab[index - 1] > random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index < difficulty - 1 && tab[index + 1] != -1 && tab[index + 1] < random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }
    }
}
```

zdjęcie 11a

```
0:
1:
2: 15
3:
4:
Wylosowana liczba to: 25
Podaj index tablicy (0 - 4): 1

Process exited with code: 0.Podano niepoprawny index. Koniec gry
```

zdjęcie 11b

Jeśli podany indeks spełnia wszystkie warunki, program umieszcza wylosowaną liczbę pod tym indeksem w tablicy (zdjęcie 12a, zdjęcie 12b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }

        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;

        if (index < 0 || index >= difficulty || tab[index] != -1) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index > 0 && tab[index - 1] != -1 && tab[index - 1] > random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index < difficulty - 1 && tab[index + 1] != -1 && tab[index + 1] < random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        tab[index] = random_number;
    }
}
```

zdjęcie 12a

```
0:
1:
2:
3:
4:
Wylosowana liczba to: 89
Podaj index tablicy (0 - 4): 3

0:
1:
2:
3: 89
4:
Wylosowana liczba to: 51
Podaj index tablicy (0 - 4):
```

zdjęcie 12b

Gra kończy się wygraną, kiedy użytkownik wpisze wszystkie wylosowane liczby w odpowiedniej kolejności. Program wyświetli wtedy komunikat: „Poprawnie wpisano wszystkie liczby. Wygrales” (zdjęcie 13a, zdjęcie 13b).

```
void Play() {
    while (usedNumbers.size() < difficulty) {
        int random_number = rand() % 101;
        while (isInUsedNumbers(random_number)) {
            random_number = rand() % 101;
        }

        usedNumbers.push_back(random_number);
        Display();
        cout << "Wylosowana liczba to: " << random_number << endl;
        cout << "Podaj index tablicy (0 - " << difficulty - 1 << "): ";
        int index = 0;
        cin >> index;

        if (index < 0 || index >= difficulty || tab[index] != -1) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index > 0 && tab[index - 1] != -1 && tab[index - 1] > random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        else if (index < difficulty - 1 && tab[index + 1] != -1 && tab[index + 1] < random_number) {
            cout << "Podano niepoprawny index. Koniec gry" << endl;
            return;
        }

        tab[index] = random_number;
    }

    Display();
    cout << "Poprawnie wpisano wszystkie liczby. Wygrales" << endl;
}
```

zdjęcie 13a

```
Process exited with code: 0.0: 13
1: 19
2: 25
3: 31
4: 44
Poprawnie wpisano wszystkie liczby. Wygrales
```

zdjęcie 13b