

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ОТЧЕТ

по лабораторной работе

на тему:

АССЕМБЛЕРНЫЕ ВСТАВКИ В ДЕЛФИ

Выполнил  
Студент гр. 951002

Проверил

В. Н. Протасеня

Асс. Е.В. Шостак

Минск, 2020

## 1. Задание на лабораторную работу

**Программа 1:** Вычисление арифметического выражения двумя способами: на языке Delphi и с помощью ассемблерной вставки. Оба варианта должны показать идентичный результат. Значения переменных вводятся непосредственно пользователем с клавиатуры. Результаты вычислений (на Delphi и с помощью ассемблерной вставки) выводятся в консоль.

**Программа 2:** Обработка одномерного *статического* массива двумя способами: на языке Delphi и с помощью ассемблерной вставки. Оба варианта должны показать идентичный результат. Размерность массива фиксирована. Массив заполняется пользователем вручную через консоль. Результаты обработки массива (на Delphi и с помощью ассемблерной вставки) выводятся в консоль.

### **Вариант 11:**

1. Вычислить значение арифметического выражения:  
$$C/(D^2 + 1) + 1$$
2. Вычислить сумму массива после минимального элемента.

## 2. Текст программы и его описание

### **Программа 1:**

## А) код на языке Delphi

```
program assembler_laba_6_part1;
{$APPTYPE CONSOLE}
{$R *.res}
uses
  System.SysUtils;
var C,D: integer;
res: real;
begin
  writeln('Enter number C: '); // вывод сообщения о просьбе ввести число C
  readln(C); // ввод числа C
  writeln('Enter number D: '); // вывод сообщения о просьбе ввести число D
  readln(D); // ввод числа D
  res:=C/(D*D+1)+1; //вычисление выражения C/(D^2+1)+1
  writeln('C/(D^2+1)+1=',res:2:0); // вывод результата вычисления
  readln;
end.
```

## Б) код с помощью ассемблерной вставки

```
program assembler_laba_6_part2;
{$APPTYPE CONSOLE}
{$R *.res}
uses
  System.SysUtils;
var C,D: word;
    result: word;
begin
  writeln('Enter number C: '); // вывод сообщения о просьбе ввести число C
  readln(C); // ввод числа C
  writeln('Enter number D: '); // вывод сообщения о просьбе ввести число D
  readln(D); // ввод числа D
  asm //начало ассемблерной вставки для вычисление выр-я C/(D^2+1)+1
    mov ax, D // кладем в регистр ax знач-е переменной D
    mul ax // D^2
    add ax,1 // D^2+1
    mov bx, ax // кладем в регистр bx знач-е вычисленного выр-я D^2+1
    xor dx, dx. // обнуление регистра dx для деления
    mov ax, C // кладем в регистр ax знач-е переменной C
    div bx // деление переменной C на (D^2+1)
    add ax, 1 // C/(D^2+1)+1
    mov result, ax // кладем в переменную result результат вычисления выр-я
  end; //конец ассемблерной вставки
  writeln('C/(D^2+1)+1=',result); // вывод результата вычисления
  readln;
end
```

Тесты к Программе 1:

А) код на языке Delphi		Б) код с помощью ассемблерной вставки	
Вводимые значения	Результат	Вводимые значения	Результат
C=8092 D=4	Enter number C: 8092 Enter number D: 4 C/(D^2+1)+1=477	C=8092 D=4	Enter number C: 8092 Enter number D: 4 C/(D^2+1)+1=477
C=30 D=2	Enter number C: 30 Enter number D: 2 C/(D^2+1)+1= 7	C=30 D=2	Enter number C: 30 Enter number D: 2 C/(D^2+1)+1=7
C=777 D=3	Enter number C: 777 Enter number D: 3 C/(D^2+1)+1=79	C=62300 D=6	Enter number C: 62300 Enter number D: 6 C/(D^2+1)+1=1684
C=8842 D=6	Enter number C: 8842 Enter number D: 6 C/(D^2+1)+1=240	C=100 D=5	Enter number C: 100 Enter number D: 5 C/(D^2+1)+1=4

## Программа 2:

### А) код на языке Delphi

```
program assembler_laba_6_2A;
{$APPTYPE CONSOLE}
{$R *.res}
uses
  System.SysUtils;
type
  MyMas= array [1..10] of word;
var Mas: MyMas;
    i: integer;
    Numb: word;
    MinIndex, Sum: word;
begin
  writeln('Enter your array: ');
  for i := 1 to 10 do
  begin
    readln(Numb);
    Mas[i]:=Numb;
  end;
  writeln;
  write('Your array: ');
  for i := 1 to 10 do
  begin
    write (Mas[i], ' ');
```

```

end;
writeln;
MinIndex:=1;
for i := 1 to 10 do
begin
  if Mas[i]<Mas[MinIndex] then
  begin
    MinIndex:=i;
  end;
end;
writeln('Minimal element: ', Mas[MinIndex]);
writeln('Minimal element's index: ', MinIndex);
Sum:=0;
for i := MinIndex+1 to 10 do
Sum:=Sum+Mas[i];
writeln('Sum of elements after minimal element: ', Sum);
readln;
end.

```

## Б) код с помощью ассемблерной вставки

```
program assembler_laba_6_2B;
{$APPTYPE CONSOLE}
{$R *.res}
uses
  System.SysUtils;
type
  MyMas= array [1..10] of Word;
var Mas: MyMas;
    i: integer;
    Numb, Sum,MinElement,MinIndex : word;
begin
  writeln('Enter your array: ');
  for i := 1 to 10 do  // ввод массива
    readln(Mas[i]);
  Writeln;
  Write('Your array: ');
  for i := 1 to 10 do  // вывод введенного массива
    Write (Mas[i], ' ');
  Sum := 0;
  writeln;
  Asm  // начало ассемблерной вставки
  //Поиск минимального элемента
  mov cx, 10  // количество элементов в массиве=количество повторений цикла
  lea edx, Mas  // вычисляем эффективный адрес, т.е полный адрес на начало нашего массива
  xor di, di  // обнуляем di, в нем будет храниться текущий индекс
```

```

xor eax, eax // Обнуляем eax
mov bx, [edx] // кладем в bx первый элемент массива,
               // в нем будет храниться значение текущего найденного минимального элемента

@@LoopSearchMin: // Цикл поиска мин. элемента
inc di // увеличиваю счетчик
cmp bx, [edx] // сравниваю текущий мин. элемент с текущим элементом в массиве
jng @@EndOfLoop // если текущий мин. элемент меньше текущего элемента массива
               // то текущий мин.элемент не меняем и идем дальше по циклу, прыгая на метку @@LoopEnd:
mov bx, [edx] // если текущий мин. элемент больше текущего элемента массива, то заменяем
mov ax, di // кладем индекс нового мин. элемента в ax

@@EndOfLoop:
add dx, 2 // здесь я обращаюсь к младшим двум байтам эффективного адреса,
           // увеличиваю его на 2 т.к. элемент массива типа word т.е 2 байта
dec cx // уменьшаю счетчик цикла
cmp cx, 0 // сравниваю cx с нулем
jne @@LoopSearchMin

mov MinIndex, ax // кладем в переменную индекс мин. элемента, для последующего вывода
// в ax индекс минимального элемента
// в edx будет эффективный адрес
// в eax будет результат

mov cx, 10 // вычисляю сколько раз должен выполняться цикл для суммирования элементов

```



```

sub cx, ax    // 10-индекс мин. элемента = кол-во повторений цикла

shl ax, 1     // умножаю ax на 2 сдвигом влево, чтобы найти смещение в массиве, умножаю на 2 т.к. каждый элемент 2 байта

lea edx, [Mas] // кладу в edx эффективный адрес

add edx, eax  // добавляем к эффективному адресу наше смещение

xor eax, eax  // обнуляю eax т.к. далее будет использоваться

@@LoopSum:   // цикл подсчета суммы элементов после минимального

add ax, [edx] // суммирую

add edx, 2    // смещаю

dec cx       // уменьшаю счетчик цикла

cmp cx, 0     // сравниваю cx с нулем

jne @@LoopSum

mov [Sum], ax // кладу в переменную Sum сумму

end; //конец ассемблерной вставки

writeln('Minimal element: ', Mas[MinIndex]);

writeln('Minimal element's index: ', MinIndex);

writeln('Sum of elements after minimal element: ', Sum);

readln;

end.

```

Тесты к Программе 2:

А) код на языке Delphi		Б) код с помощью ассемблерной вставки	
Вводимые значения	Результат	Вводимые значения	Результат
1 2 3 4 5 0 9 10 50 60	<pre> Enter your array: 1 2 3 4 5 0 9 10 50 60  Your array: 1 2 3 4 5 0 9 10 50 60 Minimal element: 0 Minimal elemet's index: 6 Sum of elements after minimal element: 129 </pre>	1 2 3 4 5 0 9 10 50 60	<pre> Enter your array: 1 2 3 4 5 0 9 10 50 60  Your array: 1 2 3 4 5 0 9 10 50 60 Minimal element: 0 Minimal elemet's index: 6 Sum of elements after minimal element: 129 </pre>

<pre> 11 12 13 14 15 6 18 17 19 20 </pre>	<pre> Enter your array: 11 12 13 14 15 6  18 17 19 20  Your array: 11 12 13 14 15 6 18 17 19 20 Minimal element: 6 Minimal element's index: 6 Sum of elements after minimal element: 74 </pre>	<pre> 11 12 13 14 15 6 18 17 19 20 </pre>	<pre> Enter your array: 11 12 13 14 15 6 18  17 19 20  Your array: 11 12 13 14 15 6 18 17 19 20 Minimal element: 6 Minimal element's index: 6 Sum of elements after minimal element: 74 </pre>
<pre> 11 677 90 54 78 43 56 33 9 22 </pre>	<pre> Enter your array: 11 677 90 54 78 43 56 33 9 22  Your array: 11 677 90 54 78 43 56 33 9 22 Minimal element: 9 Minimal element's index: 9 Sum of elements after minimal element: 22 </pre>	<pre> 189 150 20 140 30 3 56 47 5 4 </pre>	<pre> Enter your array: 189  150 20 140 30 3 56 47 5 4  Your array: 189 150 20 140 30 3 56 47 5 4 Minimal element: 3 Minimal element's index: 6 Sum of elements after minimal element: 112 </pre>

### **Вывод:**

Использование ассемблерных вставок в высокоуровневых языках, в частности Delphi, дает нам больше возможностей работы с памятью, с аппаратным обеспечением, предоставляет доступ к специфическим инструкциям процессора, появляется оптимизация кода, которая позволяет обойти ограничения компилятора.