International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 14, Number 11 (2021), pp. 1163-1168
© International Research Publication House. http://www.irphouse.com

Evaluating the Portability of array-Based Data Structures in the Pseint Language

Albert Miyer Suarez Castrillon[1], Sir-Alexci Suarez Castrillon[2] and Thomas Edison Guerrero Barbosa[3]

[1] Faculty of Engineering and Architecture, GIMUP, University of Pamplona, Colombia.

[2,3] Engineering Faculty, University Francisco of Paula Santander Ocaña, Colombia.

Abstract

The paper presents an evaluation of the programming language Pseint, in the declaration, dimensionality, size and basis of the data structure based on arrays, such as vectors and matrices. It evaluates its effectiveness as a starting program in the handling of arrays and the transition that must be made to a high-level programming language, in such a way that it allows the student an adequate assimilation of the use and concepts in the languages of Java, C++ and Phyton. In this way it can be concluded if it is suitable as a starting language in programming logic and handling of vectors and arrays.

Keywords: Pseint, matrices, vectors, arrays, data structure.

I. INTRODUCTION

Programming is a process that through instructions executed on the computer can help solve a problem, by coding in a programming language. And programming logic is necessary in almost all careers and professions today, which are still included in the curricula of programming courses [1]-[4]. Among the data that can be used are simple and compound data, the latter are classified into data structures, where one of them are the so-called one-dimensional arrays such as vectors, two-dimensional as matrices or multidimensional in arrays of three or more dimensions [5]. The use of such structures is very important for solving and handling large volumes of data. The question is whether with the programming languages that are taught in the first courses, have the full potential for teaching arrays or allows a proper transition to a high-level program [6].

In higher education institutions the initial courses are focused on the use of languages that help to understand the logic first and then use more powerful languages like Java, Phyton or C++, at the beginning programs like Scratch are very well received by students who are starting in the world of programming, where scenarios and animations that can be executed by means of blocks are integrated [7], [8]. Scratch keeps clear the concepts of variables, data types, repetitive control structures, simple and double conditional structures, with which infinite problems can be solved, however it presents deficiencies in the handling of vectors and matrices, and its programming when changing to a high-level language is not so clear for a transition, becoming a disadvantage. Logic can also be taught through flowcharts, although this application does include arrays, they have the disadvantage that you cannot define the data type, dimensionality or size, and its portability to high-level languages is not the best, on the one hand DFD is composed of symbols and not by words or reserved instructions to achieve proper coding [9].

In Latin America due to the language spoken as Spanish, there is an initial problem with programming languages which are based on the English language, this problem initially presents a barrier, which somewhat delays learning, so it creates the need to find solutions in the Spanish language, which have the full potential for the management of statement, operators, control structures and array management. At the same time that allow the transition to high-level languages with the least possible change, following the logic of the programming language.

One of the languages that presents handling characteristics in the Spanish language is Pseint, a language that allows to get started in the logic of programming while maintaining the same concepts of traditional languages [10], [11]. In this way, students focus more on logic and not on learning how to write the different sentences, which sometimes are extremely long and out of their native language. That is why many institutions start with this language and then give way to a second language in the curriculum, which can be Java, Phyton, etc., the most important thing that when using it is not an inconvenience for this transition, although it is known that Pseint can handle all control structures [12], [13], we want to know if the arrays have the necessary functionality to start in the data structures, from the point of view of declaration, dimensionality, size, base and easy handling and transition to other languages.

II. METHOLOGY

To analyze and know the potential that Pseint has in the handling of data structures called arrays and more specifically vectors and matrices, 6 phases were carried out (Figure 1), which are compared with 3 high-level languages that are generally used at the beginning of programming or after starting with the Pseint program are still in the curriculum of different courses or careers. Each of the phases is explained below.

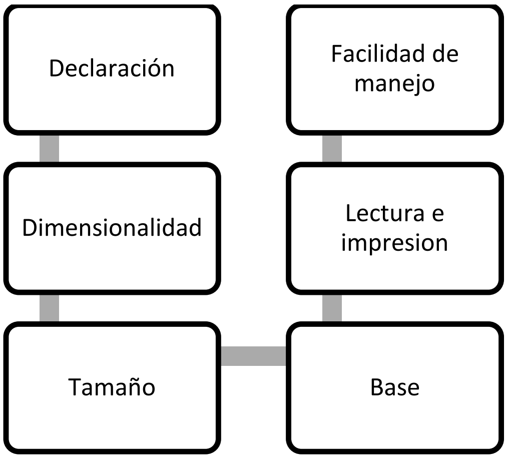


Fig. 1. Phases of analysis.

Declaration phase

The declaration is usually known as the assignment of an identifier to the name of a variable, which will contain all the stored values of the arrays. We will analyze the way to define them and if it maintains similarity with high-level languages.

Dimensionality phase

Programming languages can be one-dimensional, two-dimensional or three-dimensional, generally called cell in languages such as Matlab, allowing you to perform various mathematical operations.

Phase size

It is analyzed how many elements it can store and if it has a finite limit from the declaration or it can take values in the execution process.

Base Phase

The base is important to know because according to the high-level language that will follow in the curriculum can be adapted or should be adapted to that language without causing problems to the initial programmer.

Reading and printing phase

They are the input and output data processes, where simplicity prevails for its operation in the program.

Ease of use phase

The 5 previous items are analyzed to know its simplicity, without the way to handle vectors and matrices is complex and ends up affecting their learning or the transition to the next programming language in the curriculum.

Each of the above steps is performed by testing proposed problems or code in the Pseint language, which is compared to Java, Phyton, DFD and C languages.

III. RESULTS

Before analyzing each of the phases of the methodology a problem is presented which is proposed and then solved to determine advantages and disadvantages of the language in the 6 phases analyze, which contains the use of vectors and matrices. The first exercise focuses on vectors and the second on matrices.

Vectors

A salesperson wants to know how much bonus he has earned in his last year of sales (Table 1). Make a program that:

- Load an information vector with the sales amount for the months of the whole year for an employee.
- Create a new vector to store the bonus amount.
  According to the following conditions:
  - ➢ If the sale is over $850000, a 10% bonus is given.
  - ➢ If the sale is between $500,000 and $85,000,000, an 8% bonus is given.
  - ➢ If the sale is less than $500,000 a 5% bonus is applied to the sale.
- Print the month and bonus for each month.
- The program should print the month and the amount with the highest bonus.
- Determine and print the average annual bonus collected.

Table 1. Sales values per year.

MONTHS OF THE YEAR	SALE VALUE
January	850000
February	1500000
March	450000
April	2500000
May	3000000
June	8900000
July	5500000
August	1850000
September	9500000
October	4000000
November	8000000
December	9900000

The solution in Pseint is:

```
Algorithm sin_title
 Define Month as a character;
 Define V,Bonds as actual;
 Define C As Integer;

 Dimension Month(12),V(12), Bonds(12);
 // POINT A
    Month(1)="January";
    Month(2)="February";
    Month(3)="March ";
    Month(4)="April ";
    Month(5)="May ";
    Month(6)="June ";
    Month(7)="July";
    Month(8)="August";
    Month(9)="September";
    Month(10)="October";
    Month(11)="November";
    Month(12)="December";
 for C=1 to 12 do
    Type "Enter the sales value for the month of ", Month(C);
    Read V(C);
 FinFor

 //POINT B
 for c=1 to 12 Make
    if V(C)> 850000 Then
       Bonds(C)=V(C)*0.1;
    YesNo
       if V(C)≥ 500000 and V(c)≤850000 then
          Bonds(c)=V(c)*0.08;
       YesNo
          if V(c)≥0 and V(C)< 500000 Then
             Bonds(c)=V(c)*0.05;
          YesNo
             Bonds(C)=0;
          FinSi
       FinSi
    FinSi
 FinFor


 //DOT C
 For c=1 to 12 Make
    Type "The bonus of the month of ",Month(C);
    Type Bonds(C);
 FinFor

EndAlgorithm
```

From the previous solution it can be analyzed that it maintains the possibility of declaration, dimensionality, size assignment, creation or filling of the vector manually through the respective index and use of repetitive structures when you want to enter the values of each index of the vector by keyboard or even randomly depending on the problem to be solved.

Matrices

Given the matrix in Table 2, do the following:

- Declare, size and create an array.
- Print the largest value and its position.
- Print the average of each row.
- Print the average of each row.
- Print the average of the whole matrix.
- Find a keyed value in the array.

Table 2. Matrix value exercise 2.

	1	2	3	4	5
1	3	4	20	5	80
2	10	22	75	23	43
3	28	48	12	65	2
4	0	60	3	79	33

The proposed solution is as follows:

```
Algorithm sin_title
 // MATRIXES
 define A, F, C,may,posf,posc,sf,SC, S,V as integer;
 define p as real;
 dimension A[4,5];
 // filling the matrix
 for f←1 to 4 Make
    for c←1 to 5 Make
       Type "enter position value ",F,",",C;
       read A[F,C];
    FinFor
 FinFor
 //Search for the highest value
 may=A[1,1];
 posf=1;
 posc=1;
 for f←1 up to 4 do
    for c←1 to 5 Make
       if A[f,c]> may then
          may=A[f,c];
          posf=f;
          posc=c;
       FinSi
    FinFor
 FinFor
 Write "The largest value is ",may;
 Write "Row ",posf;
 Write "Column ", posc;
 // average of each row
 for f←1 up to 4 do
    SF=0;
    for c←1 up to 5 do
       SF=SF+A[f,c];
    FinFor
    P=SF/5;
    Type "Row average ",F;
    Type P;
 FinFor
 // average of each column
 for c←1 up to 5 do
    SC=0;
    for f←1 up to 4 do
       SC=SC+A[f,c];
    FinFor
    P=SC/4;
    type "Average column ",C;
    Type P;
 FinFor
 //average the whole matrix
 S=0;
 for f←1 up to 4 do
    for c←1 up to 5 do
       S=S+A[f,c];
    FinFor
 FinFor
```

```
  P=S/20;
  Write "The average of the whole matrix is ",P;
  //average of column 4
  sc=0;
  for f←1 up to 4 do
    sc=sc+A[f,4];
  FinFor
  p=sc/4;
  type "Average column 4",p;
  //Find a value
  type "Enter value to search for ";
  read V;

  for f←1 up to 4 do
    for c←1 up to 5 do
      if V=A[f,c] Then
        Write "value found";
      FinSi
    FinFor
  FinFor
EndAlgorithm
```

After showing two exercises with vectors and matrices you can get a clear idea of the functionality of the language for each result in the analysis phases:

In the declaration phase, the definition of the data types that will store the vectors and arrays are similar to high-level languages (Table 3), which is a benefit in the transition to new languages, only the unique way of writing the commands in each language must be changed, where the name of the reserved words for the string and real types changes, also integers and logical types can be declared.

Table 3. Comparison of phase 1 with high-level languages.

Language	Statement
Pseint	Define bonds as actual; Define V as actual; Define Month as a character;

Java	double bonds[];
	double v[];
	String month[];

C++	float bonds[12];
	string month[12];
	float v[12];

Python	bonds = float()
	month = str()
	v = float()

In table 4, you can see that the dimensionality in Pseint can only work with one-dimensional vectors and two-dimensional matrices, it does not allow the use of multidimensional work, for example 3 dimensions, however it is still quite efficient and powerful to work with arrays.

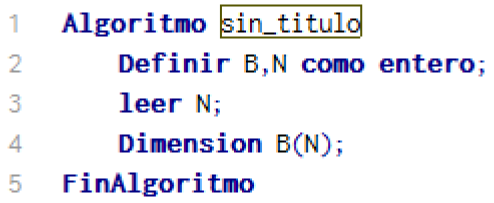
Table 4. Comparison of phase 2 with high-level languages.

Language	Dimensionality		
	One-dimensional	**Two-dimensional**	**Three-dimensional**
Pseint	X	X	
Java	X	X	X
C++	X	X	X
Python	They function as lists		

In table 5, the size or space reserved for the creation of the matrix A, which has values with 4 rows and 5 columns, is similar to other languages. The arrays are data of the same type stored in cells and which by means of the index can be accessed, but the size must be finite, but sometimes it is not clear in the source code the size of the same therefore can be entered in the execution of the program, this advantage also has the Pseint language.

Table 5. Comparison of phase 3 with high-level languages.

Language	Statement
Pseint	Dimension A(4,5);
Java	a = new int[4][5];
C++	int a[4][5];
Python	It works as lists, but can be placed as: a = [[int() for ind0 in range(5)] for ind1 in range(4)]



```
Algoritmo sin_titulo
    Definir B,N como entero;
    leer N;
    Dimension B(N);
FinAlgoritmo
```

Fig. 2. Assigning running size to a vector.

In table 6, you can see the base of each language. The advantage of Pseint is that it can be configured to start at 0 or 1 (table 7 and 8), by starting at 1 it is easier for the student who starts programming and can easily adapt to the base of the next language in the curriculum, because he already knows the concept of vectors and matrices.

Table 6. Comparison of phase 4 with high-level languages.

Language	Base	
	0	1
Pseint	X	X
Java	X	
C++	X	
Python	X	

Table 7. Base vector from 1 in Pseint

Index	1	2	3	4	5	6	7
Value	3	4	20	5	80	24	4

Table 8. Base vector from 0 in Pseint

Index	0	1	2	3	4	5	6
Value	3	4	20	5	80	24	4

In table 9, the reading and writing of the first exercise is analyzed. Where the advantage of Pseint for writing and reading data is clearly evident, very similar to the ease of Python and languages such as Matlab, making it understandable for the student who is just starting out in programming. It can be observed that java presents complex reserved words to perform two of the most common tasks in programming.

Table 9. Comparison of phase 5 with high-level languages.

Language	Statement
Pseint	Type "Enter the sales value for the month of ", Month(C); Read V(C);
Java	System.out.println("Enter the sales value for the month of "+month[c-1]); v[c-1] = Double.parseDouble(bufInput.readLine());
C++	cout << "Enter the sales value for the month of " << month[c-1] << endl; cin >> v[c-1];
Python	print("Enter the sales value for the month of ",month[c-1]) v[c-1] = float(input())

The analysis of phase 6 shows that Pseint is an easy-to-use language for matrices and vectors, as well as for the entry and printing on screen of the data of each array, maintaining similarities with the languages of initiation in programming, where the natural language of Spanish is maintained. In addition, the configuration can be adapted from the beginning of the course to the next high-level language in the curriculum.

IV. CONCLUSION

The programming language Pseint, allows to start in the world of logic programming in a simple and friendly way in the Spanish language, with which you can quickly start making the first programs, keeping all repetitive and conditional control structures, you can also see the potential in the management of vectors and matrices, allowing the student to make the transition to the next high-level language in the curriculum, where you should only interpret how to write the words reserved for the management of data structures based on arrays.

REFERENCES

[1] M. V. Rosas, M. E. Zúñiga, J. Fernández, and R. A. Guerrero, "Computational thinking at the university level," presented at XIX Workshop of Researchers in Computer Science (WICC 2017, ITBA, Buenos Aires), Sep. 2017. Accessed: Nov. 25, 2021. [Online]. Available at: http://sedici.unlp.edu.ar/handle/10915/62319

[2] "Should programming be taught in all college majors?" https://www.xataka.com/otros/debe-ensenarse-programacion-en-todas-las-carreras-de-la-universidad (accessed Nov. 25, 2021).

[3] G. E. Chanchí, P. Á. Sánchez, and W. Y. C. Muñoz, "Strategies for teaching programming logic in engineering," *Encuentro Int. Educ. En Ing.* , Aug. 2018, Accessed: Nov. 25, 2021. [Online]. Available at: https://acofipapers.org/index.php/eiei/article/view/360

[4] V. S. D'Angelo, "Computer programming. Reflections on the need for an interdisciplinary approach", *Rev. Iberoam. Cienc. Tecnol. Soc. - STS*, vol. 13, n. °39, pp. 111-141, 2018.

[5] "Educational software for working with matrices | Digital Journal: Mathematics, Education and the Internet", Jun. 2016, Accessed: Nov. 25, 2021. [Online]. Available at: https://revistas.tec.ac.cr/index.php/matematica/article/view/2525

[6] L. N. Bautista, R. E. C. Valencia, and A. F. Morales, "Student engineering tool: matrix calculator," *Rev. Links*, vol. 13, n. °1, Art. n. °1, Jun. 2016, doi: 10.14483/2322939X.11660.

[7] C. L. Vidal, C. Cabezas, J. H. Parra, and L. P. López, "Practical Experiences with the Use of the Scratch Programming Language to Develop the Algorithmic Thinking of Students in Chile," *Form. Univ.* , vol. 8, n. °4, pp. 23-32, 2015, doi: 10.4067/S0718-50062015000400004.

[8] M. Mendoza Aguirre, "Programming Software "Scratch" in the Development of Mathematical Logical Thinking of Students of a Primary Educational Institution, Chincha - 2017", *Univ. Cesar Vallejo*, 2018, Accessed: Nov. 25, 2021. [Online]. Available at: https://repositorio.ucv.edu.pe/handle/20.500.12692/29889

[9] D. Y. Valentin Lopez and Y. ysidora Zambrano Fernandez, "Influencia del software libre DFD en el área de educación para el trabajo para un aprendizaje significativo, en los alumnos del 5to "B" de la Institución Educativa Daniel Alcides Carrión de Cerro de Pasco-2017", *Univ. Nac. Daniel Alcides Carrión*, oct. 2019, Accessed: nov. 25, 2021. [Online]. Available at: http://repositorio.undac.edu.pe/handle/undac/1597

[10] N. Cabanes, *Introduction to pseudocode programming with PSeInt*.

[11] J. E. Beúnes Cañete, A. Vargas Ricardo, J. E. Beúnes Cañete, and A. Vargas Ricardo, "The introduction of the didactic tool PSeInt in the teaching-learning process: a proposal for Linear Algebra," *Transformation*, vol. 15, n. º1, pp. 147-157, Apr. 2019.

[12] I. K. Rincon, S.-A. Suarez, and A. M. Suarez, "Evaluation of the pseudo language Pseint for its high level language characteristics", p. 6.

[13] F. H. Evangelista and P. J. Novara, "Interpreter for testing a program written in pseudocode," *Ind. Data*, vol. 17, n. º1, Art. n. º1, Jun. 2014, doi: 10.15381/idata.v17i1.12039.