*Proceedings of the $2^{nd}$ IMT-GT Regional Conference on Mathematics, Statistics and Applications*
*Universiti Sains Malaysia*

# A VISUAL MODEL FOR COMPUTING SOME PROPERTIES OF $U(n)$ AND $\mathbb{Z}_n$

Nor Muhainiah Mohd Ali, Deborah Lim Shin Fei, Nor Haniza Sarmin, Shaharuddin Salleh
Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 Skudai, Johor Darul Takzim
muhainiah9119@yahoo.com, debra_shinfei@yahoo.com, nhs@mel.fs.utm.my, ss@mel.fs.utm.my

**Abstract.** A computer program is developed using Microsoft Visual C++ in the Windows environment. This program focuses on two specific finite Abelian groups, which are the group $\mathbb{Z}_n$ under addition modulo $n$ and the group $U(n)$ under multiplication modulo $n$, where $n$ is any positive integer less than or equal to 120. Computations of the properties of the two groups get more tedious and time consuming as the value of $n$ increases. Therefore a program that could assist in the computation would indeed be of great help. This program in C++ is written relating to some properties of $\mathbb{Z}_n$ and $U(n)$. It enables the user to enter any positive integer $n$ ($n \leq 120$) to generate answers to some properties of these two groups. A lattice diagram can be obtained for any groups of $\mathbb{Z}_n$ and, for groups of $U(n)$ which are cyclic.

## 1. Introduction

The term **group** was coined by Galois about 160 years ago to describe sets of one-to-one functions on finite sets that could be grouped together to form a closed set [1]. Similar to the case of most fundamental concepts in mathematics, the modern definition of a group that will be given in the following section comes from a long evolutionary process. This definition was given by both Heinrich Weber and Walther von Dyck in 1882 [1].

C++, an extension of C, was developed by Bjarne Stroustrup in the early 1980s at Bell laboratories [4]. C++ provides capabilities for object-oriented programming and this is among the attractiveness of this programming language, as object-oriented programs are easier to understand, correct and modify.

Working on Microsoft Visual C++ in the Windows environment gives the programmer the ability to simulate as well as to visualize the problem that one is doing. Working in such environment also helps the users to have better visualization of the problem, and the programs are more user-friendly.

Therefore, Microsoft Visual C++ in the Windows environment is used to simulate some properties of the groups $\mathbb{Z}_n$ and $U(n)$ and to visualize the lattice diagram. This program enables the user to enter the desired value of $n$, for $n \leq 120$. Most properties displayed in the interface can be then obtained in the text file. The purpose is to serve as a convenience for the user to print out the output.

Currently in the market, there are two popular softwares which are able to solve problems in group theory. They are Groups, Algorithms and Programming (GAP) and the Magma Computational Algebra System. However, there is not yet any available software in the market which is user-friendly and, that is able to solve problems and finding properties of any groups written using Microsoft Visual C++ in the Windows environment. Therefore, this program serves as a starting point for developing better programs using Microsoft Visual C++ in the Windows environment.

## 2. The groups $\mathbb{Z}_n$ and $U(n)$

Some related definitions on groups, properties of groups, as well as explanation on how to obtain some properties of $\mathbb{Z}_n$ and$U(n)$ are included in this section.

**Definition 2.1.** [2] *A **binary operation** $*$ on $S$ is a function mapping $S \times S$ into $S$. For each $(a,\ b) \in S \times S$, we will denote the $*((a,\ b))$ of $S$ by $a * b$.*

For simplification, $a * b$ can just be written as *ab*.

**Definition 2.2.** [1] *Let G be a nonempty set together with a binary operation (usually called multiplication) that assigns to each ordered pair (a, b) of elements of $G \times G$ an element in G denoted by ab. We say G is a **group** under this operation if the following three properties are satisfied:*

(1) *Associativity. The operation is associative; that is (ab)c = a(bc) for all a, b, c in G.*
(2) *Identity. There is an element $e$ (called the identity) in G, such that ae = ea = a for all $a$ in G.*

(3) *Inverses. For each element $a$ in G, there is an element $b$ in G (called the inverse of $a$) such that ab = ba = e.*

*A group is **Abelian** if the group has the property of ab = ba for every pair of elements $a$ and $b$. In short, this means that the group is commutative. Therefore, a group is **non-Abelian** if there is some pair of elements $a$ and $b$ for which $ab \neq ba$.*

**Definition 2.3.** [2] *The number of elements of a group (finite or infinite) is called its **order**. The notation, $|G|$ will be used to denote the order of G.*

**Definition 2.4.** [1] *The **order of an element** g in a group G is the smallest positive integer $n$ such that $g^n = e$ (In additive notation, this would be $ng = 0$). The order of an element g is denoted by $|g|$.*

**Definition 2.5.** [1] *If a subset $H$ of a group $G$ is itself a group under the operation of G, we say $H$ is a **subgroup** of G.*

**Definition 2.6.** [1] *A group G is called **cyclic** if there is an element $a$ in G such that $G = \{a^n | n \in \mathbb{Z}\}$. Such an element $a$ is called a generator of G.*

**Definition 2.7.** [3] *Let $a \in G$. Then $\langle a \rangle = \{a^n\, | n \in \mathbb{Z}\} = \{e, a, a^2, a^3, ...\}$ is called a **cyclic subgroup** of G generated by a.*

**Definition 2.8.** [3] *This diagram is drawn to show the subgroups of a group. In such a diagram, a line running downward from a group G to a group $H$ means that $H$ is a subgroup of G. Thus the larger group is placed nearer to the top of the diagram.*

**Definition 2.9.** [3] *The set $\mathbb{Z}_n = \{0, 1, 2, ..., n-1\}$ for $n \geq 1$ is a group under addition modulo $n$. For any $i$ in $\mathbb{Z}_n$, the inverse of i is n – i. This group is usually referred to as **the group of integers modulo n**.*

The following is an example of a group $\mathbb{Z}_n$ that is $\mathbb{Z}_4$ under addition modulo 4 with some of its properties.

**Example 2.1.** The elements $\mathbb{Z}_4$ are $0,\ 1,\ 2$ and $3$. Hence the order of the group is $4$. The computations of the order of the elements are as follows:

$|0| = 1$ since the order of the identity element is always $1$.

$|1| = 4$ since $1 + 1 + 1 + 1 \equiv 0$.

$|2| = 2$ since $2 + 2 \equiv 0$.

$|3| = 4$ since $3 + 3 + 3 + 3 \equiv 0$.

One way of getting the inverse of each element is we can use the formula $n – i$, where $i$ is the element of $\mathbb{Z}_4$. Therefore, $0^{-1} = 0,\quad 1^{-1} = 3,\ 2^{-1} = 2$, and finally $3^{-1} = 1$. The generators of this group are 1 and 3 since the order of these elements are the same as the order of the group. The cyclic subgroups of $\mathbb{Z}_4$ are obtained by generating each element of the group. The following shows the cyclic subgroups of $\mathbb{Z}_4$:

$\langle 0 \rangle = \{0\},\quad \langle 1 \rangle = \{0,\ 1,\ 2,\ 3\},\quad \langle 2 \rangle = \{0,\ 2\}$, and $\langle 3 \rangle = \{0,\ 1,\ 2,\ 3\}$.

Hence the lattice diagram of $\mathbb{Z}_4$ is:

$$\begin{tikzcd}
\mathbb{Z}_4 = \langle 1 \rangle = \langle 3 \rangle \arrow[d, dash] \\
\langle 2 \rangle \arrow[d, dash] \\
\langle 0 \rangle
\end{tikzcd}$$

**Definition 2.10.** [3] *For each $n > 1$, we define $U(n)$ to be the set of all positive integers less than $n$ and relatively prime to $n$. Then $U(n)$ is **a group under multiplication modulo n**.*

In the following is an example of a group $U(n)$, that is $U(5)$ under multiplication modulo 5 and some of its properties.

**Example 2.2.** The elements of $U(5)$ consists of $1,\ 2,\ 3,$ and $4$. Hence the order of the group is $4$. The computations of the order of the elements are as follows:

$|1| = 1$ since the order of the identity element is always $1$.

$|2| = 4$ since $2 \times 2 \times 2 \times 2 \equiv 1$.

$|3| = 4$ since $3 \times 3 \times 3 \times 3 \equiv 1$.

$|4| = 2$ since $4 \times 4 \equiv 1$.

The inverse of each elements are: $1^{-1} = 1, \quad 2^{-1} = 3, \quad 3^{-1} = 2$, and finally $4^{-1} = 4$. To find the inverse, say $2^{-1}$, we compute powers of 2 until we get the identity, that is $2 \times 1 \equiv 2,\ 2 \times 2 \equiv 4,\ 4 \times 2 \equiv 3,\ 3 \times 2 \equiv 1$. Therefore $2^{-1} = 3$. Generators of this group are 2 and 3. The cyclic subgroups of $U(5)$ are also obtained by generating each element of the group. The following shows the cyclic subgroups:

$\langle 1 \rangle = \{1\}, \quad \langle 2 \rangle = \{1,\ 2,\ 3,\ 4\}, \quad \langle 3 \rangle = \{1,\ 2,\ 3,\ 4\}$, and $\langle 4 \rangle = \{1,\ 4\}$.

Hence the lattice diagram of $U(5)$ is as follows of $U(5)$.

$$\begin{tikzcd}
U(5) = \langle 2 \rangle = \langle 3 \rangle \arrow[d, dash] \\
\langle 4 \rangle \arrow[d, dash] \\
\langle 1 \rangle
\end{tikzcd}$$

## 3. Some programming codes

We include in this paper some codes written for the program.

**3.1. Some important codes.** The following codes are to create a small box (called edit box) for the users to enter the value of $n$, and to create a place for the users to select their desired groups (called radio buttons), as well as to create a button that will start the computation of the properties for the selected group.

```
enterN.Create(WS_CHILD | WS_VISIBLE | WS_BORDER | SS_CENTER,
CRect(CPoint(55,55),CSize(50,25)), this, IDC_ENTER_VALUE);

btnGrpG.Create("Groups", WS_CHILD | WS_VISIBLE | BS_GROUPBOX |
WS_GROUP,CRect(CPoint(10,95), CSize(120,90)), this,
IDC_BTNGRP_GROUP);

btnOptZ.Create("Z(n)", WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON,
CRect(CPoint(15,125), CSize(100,30)), this, IDC_BTNOPT_Z);

btnOptU.Create("U(n)", WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON,
CRect(CPoint(15,145), CSize(100,30)), this, IDC_BTNOPT_U);

btnCalc.Create ("OK!", WS_CHILD | WS_VISIBLE | BS_DEFPUSHBUTTON,
CRect(CPoint(150,120),CSize(50,25)), this, IDC_CALC);
```

Below are the codes for generating positive integers relatively prime to $n$ for the groups $U(n)$.

```
int CMainFrame::gcd(int x, int y)
{
    if (y==0)
        return x;
    else
        gcd(y, x%y);
}
```

Next are codes written to enable most output in the interface to be displayed in the text file. The properties that are displayed in the text file are the order of the group, the elements, generators and cyclic subgroups (if any), as well as inverse and the order of the elements. The purpose of enabling the output to be displayed in the text file is to allow the users to save the mentioned properties for the group that they have selected. Furthermore, these properties can also be printed out when displayed in the text file.

```
void CMainFrame::save_file(int input, char g, int *el, int *gen, int
*subGen, int OrderGroup, int numGenGroup, int numGenSub) {   int i,
q=0;
    ofstream output;
    output.open("Output.txt");
output << "Order of the group " << g << "(" << input << ") = " <<
OrderGroup << endl; if(g=='U')
    {
        q=1;
        OrderGroup++;
    }
output << endl << endl << "Elements of " << g << "(" << input <<
"):" << endl;
        for(i=q; i<OrderGroup; i++)
            output<< el[i] << "\t";
output << endl << endl << endl <<"Generators for "<< g << "(" <<
input << "):  (if any)" << endl;
        for(i=1; i<=numGenGroup; i++)
            output<< gen[i] << "\t";
output << endl << endl << endl << "Generators of cyclic subgroups:
(if any)" << endl;
        for(i=1; i<=numGenSub; i++)
            output<< subGen[i] << "\t";
output<< endl << endl << endl << "ELEMENTS:" << "\t" << "INVERSE:"
<< "\t" << "ORDER:" << endl;
    if(q==1)
    {
        for(i=q; i<OrderGroup; i++)
output << el[i] << "\t\t" << inv[i] << "\t\t" << order[i] << endl;
    }
    else
    {
        for(i=0; i<OrderGroup; i++)
output << el[i] << "\t\t" << (N-a[i])%N << "\t\t" << Sum[i] << endl;
    }
    output.close();
}
```

## 4. The output

This section shows how the interface of the program looks like and also displays some outputs.

**4.1. Interface.** The figure below shows the interface for the written program.

**4.2. Some outputs.** Figure 2 shows the outputs displayed in the interface when the user keys in $n = 18$ and selects the group under addition modulo $n$, meaning that $\mathbb{Z}_{18}$ is selected.

The outputs on the group $U(81)$ is shown in Figure 3.

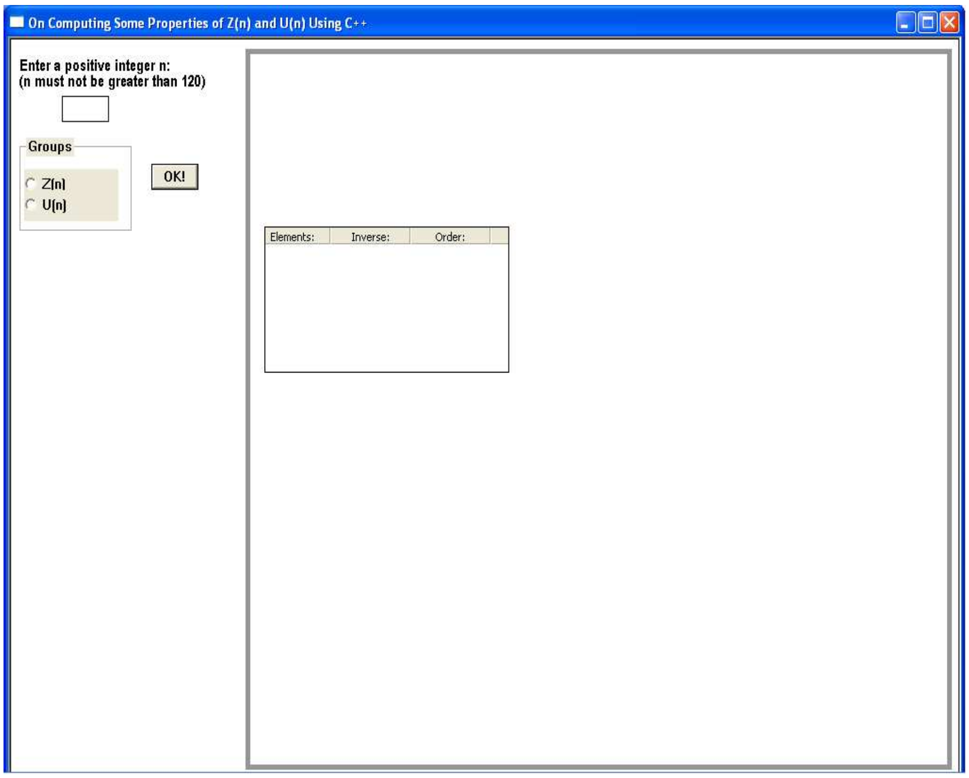
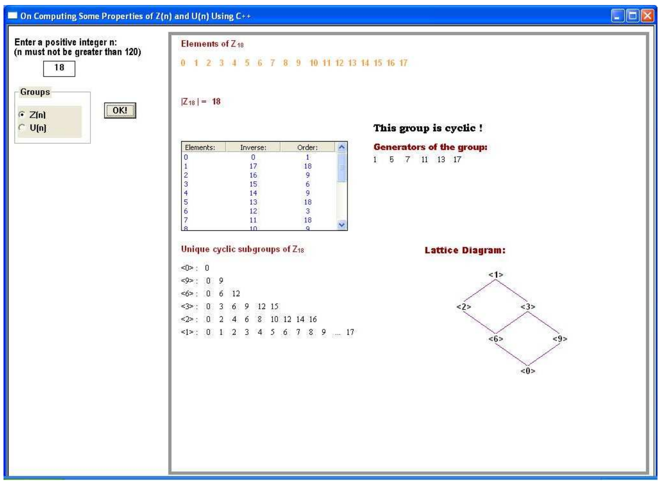

On Computing Some Properties of Z(n) and U(n) Using C++
Enter a positive integer n:
(n must not be greater than 120)
Groups
Z[n]
U[n]
OK!
Elements:
Inverse:
Order:

Figure 1


On Computing Some Properties of Z(n) and U(n) Using C++
Enter a positive integer n:
(n must not be greater than 120)
Groups
Z[n]
U[n]
OK!
Lattice Diagram:

18

[✓ Z[n]] U[n]

Elements of $Z_{18}$

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

$|Z_{18}|$ = 18

This group is cyclic !

Generators of the group:

1 5 7 11 13 17

| Elements: | Inverse: | Order: |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 17 | 18 |
| 2 | 16 | 9 |
| 3 | 15 | 6 |
| 4 | 14 | 9 |
| 5 | 13 | 18 |
| 6 | 12 | 3 |
| 7 | 11 | 18 |
| 8 | 10 | 9 |

Unique cyclic subgroups of $Z_{18}$

```
<0> :  0
<9> :  0  9
<6> :  0  6  12
<3> :  0  3  6  9  12  15
<2> :  0  2  4  6  8  10  12  14  16
<1> :  0  1  2  3  4  5  6  7  8  9  ....  17
```

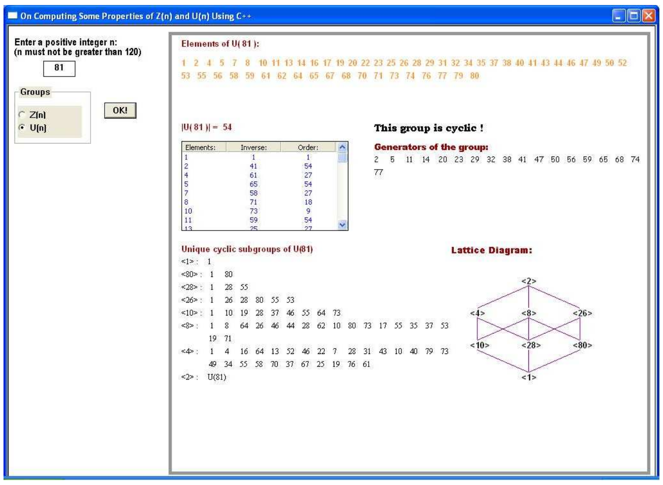
$$\begin{tikzcd}
 & <1> \arrow[dl, no head] \arrow[dr, no head] & \\
<2> \arrow[dr, no head] & & <3> \arrow[dl, no head] \arrow[dr, no head] & \\
 & <6> \arrow[dr, no head] & & <9> \arrow[dl, no head] \\
 & & <0> &
\end{tikzcd}$$

Figure 2

As mentioned, some outputs displayed in the interface can also be obtained from the text file, as shown in Figure 4.

**4.3. Message boxes.** This program is written with message boxes that will appear in different situations. One of the message boxes is used to warn the user when the user enters values of $n$ less than or equal to 0 and values of $n$ greater than 120, meaning that the value entered is out of range. The following figure shows how this message box looks like.

Another message box (see Figure 4.6) informs the user about the display, that is, cyclic subgroups and lattice diagram will only be displayed if the selected group is cyclic.



Enter a positive integer n:
(n must not be greater than 120)

81

( ) Z(n)
[✓ U(n)]

Elements of U( 81 ):

1 2 4 5 7 8 10 11 13 14 16 17 19 20 22 23 25 26 28 29 31 32 34 35 37 38 40 41 43 44 46 47 49 50 52
53 55 56 58 59 61 62 64 65 67 68 70 71 73 74 76 77 79 80

|U( 81 )| = 54

| Elements: | Inverse: | Order: |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 41 | 54 |
| 4 | 61 | 27 |
| 5 | 65 | 54 |
| 7 | 58 | 27 |
| 8 | 71 | 18 |
| 10 | 73 | 9 |
| 11 | 59 | 54 |
| 13 | 25 | 27 |

This group is cyclic !

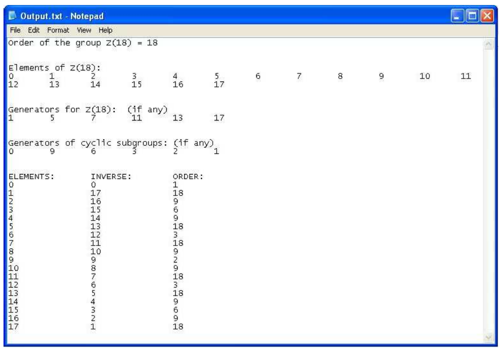
Generators of the group:

2 5 11 14 20 23 29 32 38 41 47 50 56 59 65 68 74
77

Unique cyclic subgroups of U(81)

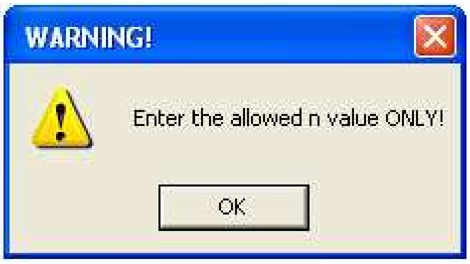
```
<1> :  1
<80> : 1  80
<28> : 1  28  55
<26> : 1  26  28  80  55  53
<10> : 1  10  19  28  37  46  55  64  73
<8> :  1  8   64  26  46  44  28  62  10  80  73  17  55  35  37  53
       19 71
<4> :  1  4   16  64  13  52  46  22  7   28  31  43  10  40  79  73
       49 34  55  58  70  37  67  25  19  76  61
<2> :  U(81)
```

Figure 3



```
Order of the group Z(18) = 18


Elements of Z(18):
0       1       2       3       4       5       6       7       8       9       10      11
12      13      14      15      16      17


Generators for Z(18):  (if any)
1       5       7       11      13      17


Generators of cyclic subgroups: (if any)
0       9       6       3       2       1


ELEMENTS:           INVERSE:            ORDER:
0                   0                   1
1                   17                  18
2                   16                  9
3                   15                  6
4                   14                  9
5                   13                  18
6                   12                  3
7                   11                  18
8                   10                  9
9                   9                   2
10                  8                   9
11                  7                   18
12                  6                   3
13                  5                   18
14                  4                   9
15                  3                   6
16                  2                   9
17                  1                   18
```

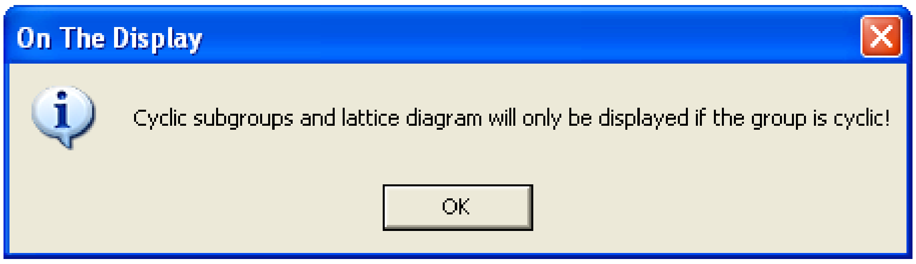
Figure 4



WARNING!

Enter the allowed n value ONLY!

Figure 5



On The Display

Cyclic subgroups and lattice diagram will only be displayed if the group is cyclic!

Figure 6

## 5. Conclusion

This program can be used to determine all elements of the group, order of the group, inverse and order of the elements, generators of the group, cyclic subgroups as well as the lattice diagram, if any, for the groups $\mathbb{Z}_n$ and$U(n)$ that are cyclic. By just entering the desired value of $n$ and selecting a group, a click of the button "OK!" will display the properties of the group.

The writing of this program is hoped to be able to facilitate in the study of group theory for the groups $\mathbb{Z}_n$ and$U(n)$ in the undergraduate level of tertiary studies. Although the program may not be able to provide with all the properties of these two groups, it is served as a starting point for developing better and sophisticated programs.

## 6. Suggestions

The input value of $n$ for this program is limited to positive values of $n$ up to 120. Therefore our suggestions would be eliminating this restriction on the value of $n$. This program can also be improved by letting the user to specifically choose the desired property to be displayed one by one, instead of having everything displayed in the interface all at once.

Apart from that, an improved program to find other cyclic subgroups of $U(n)$ for non-cyclic groups can also be developed. Furthermore, this program can also be extended to find some properties of other groups such as permutation groups and quaternion groups.

## References

[1] Gallian, J. A. *Contemporary Abstract Algebra*, $3^{rd}$ ed. D. C. Heath and Company, Canada, 1994

[2] Fraleigh, J. B. *Abstract Algebra*, $6^{th}$ ed. Addison-Wesley Publishing Company, Inc. United States of America, 2000

[3] Sarmin, N. H. *Lecture Notes: Modern Algebra SSM4113*, Preliminary Edition, Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, Malaysia, 2005

[4] Deitel, H. M and Deitel, P. J. *C++ How to Program*, $2^{nd}$ ed., Prentice Hall, United States of America, 1998

[5] Jones, R. M. *Introduction to MFC Programming with Visual C++*, Prentice Hall, United States of America.

[6] Salleh, S. *SSM3323 Lecture Notes: C to C++ Transition*, Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, Malaysia.

[7] Salleh, S. *SSM3323 Lecture Notes: C++ Programming with MFC*, Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, Malaysia.