

ASSIGNMENT
Implement a QR payment support tool
(Version v1.1)

1. Introduction

Nowadays, the payment trend in the economy is shifting towards using methods of non-cash payment. These methods bring many advantages to bank users:



- **No need to keep large amounts of cash:** Non-cash payment users will not be anxious about being robbed, counterfeit money or fire incidents.
- **No need to meet face to face:** Cashless payment users will save travel time to meet, thereby enhancing their productivity or giving them more free time.
- **Transaction information is stored:** When using cash payment, people who need to manage expenses casually have to record their spending on paper forms like receipts or notes. This way of expenses management may be hard to track and time-consuming. On the other hand, cashless payment stores all people's transactions in detail and automatically. Thereby, the need to review their transactions can be conducted at any time.
- **Get interest from the bank:** Idle money in the account can be used to save money in the bank and bring extra income (interest) to the user.



Realizing the above benefits, banks began to race with technologies to support cashless payment. E-banking and mobile banking are the iconic services that initiate the movement of cashless payment. In these services, the payment can be established by entering the recipient's account number and bank name. Afterward, the bank system will check the above information and confirm whether the input is matched with the correct person/account in the bank. If the input is confirmed, the remitter continues to enter the amount of money to transfer and the message (if any) to the recipient.

Thanks to the National Payment Gateway Service from NAPAS¹, all interbank transactions are instantaneous, just like paying cash. However, when it comes to in-person payments (such as payments at cashier counters in stores or supermarkets), online payments still have its own limitations. First of all, the sender needs to know the recipient's correct account number, which is often a long and difficult number to remember. In addition, the sender can enter this number incorrectly, which results in laborious efforts for the sender to retry. In some cases, if the sender is careless, his/her money can be sent to wrong person!

Perhaps, there still exists a solution that can further improve the above method in terms of interaction. This solution also allows users to make interbank, cashless payments easier, faster and with less error user input. And now that solution can be achieved by using QR code scanning application, so called QR payment.

¹ [The National Payment Corporation of Vietnam - NAPAS](#)



The QR code payment solution is mostly implemented as follows:

1. The seller/cashier will generate a QR code containing his/her account information and give it to the sender.
2. The customer at checkout will use their smartphones to scan the above QR code and make a money transfer on their Mobile Banking application.



Smartphone nowadays is an indispensable item for every human being. Therefore, QR payment is a solution that may replace cash payment method.

This solution can be applied to small convenience stores when dealing with customers and minimizing contact. Subsequently, it is possible to reduce the risk of spreading the epidemic of the current COVID-19 pandemic. At the sametime, for people, especially in Ho Chi Minh City and southern provinces, shopping and payment will take place in a safer way. The distribution environment will not be congested, difficult; and reduce the overall economic loss of the localities.

Within the scope of this Assignment, each student needs to implement a tool to support QR payments, more specifically, interbank payments. This tool will read the QR codes of given banks (the format of will also be given) and return the information of recipient as well as allow interbank transfers.

2. Requirements

In this Assignment, student needs to implement a C program to simulate a simple QR payment support tool. In particular, this program will read some user inputs (number or string) from the keyboard, which is the message of QR code. Then, the program will process and print to the screen the results obtained from the execution process.



3. Program Specifications

The input data will consist of many lines, in which the first line is used to identify the function (**Menu_code**). The following lines are values that are used to run that function

The format of the first line will look like this:

Menu_code
<Line 2>
...

The **Menu_code** parameter is used to identify the functionality of the QR payment support tool, namely:

- If **Menu_code** is 1: View recipient's information from QR code message
- If **Menu_code** is 2: Convert format of the QR code message
- If **Menu_code** is 3: Encrypt the message of QR code

4. QR code message format of banks

In this big exercise, students need to understand the format of the bank's QR messages to be able to process them. Knowing that, the information in the message of the QR includes:

- **Prefix numbers (of Vietnam account)**: Always the numeric string "00020101021".
- **Account number**: A string full of numeric characters, ranging from 8 to 15 characters depending on the bank.
- **Account holder name**: String containing the account holder's name, consisting of uppercase and lowercase characters (without spaces), up to 30 characters in length.
- **Bank name**: Can be one of the following three strings: "BKB", "KHB", "HBB"
- **Creation time**: String containing the time of QR creation, depending on the bank, up to 8 characters in length.

From the above information, each bank stipulates a QR code message for their own bank. The details of these messages will be presented in below subsections.

4.1. Bach Khoa Bank (BKB)

Bach Khoa Bank (BKB) defines their QR message as a sequence of consecutive information fields as follows:



Prefix numbers**Bank name****Account number****Account holder name****Creation time**

In which:

- **Bank name:** “BKB”.
- **Account number:** Is a string of numeric characters with 13 characters in length.
- **Creation time:** Is a string of days, months and years in the format **ddmmyyyy**.
Example: 22/08/2021 will be stored as “22082021”.

Some examples for BKB's QR code message:

- 00020101021BKB0271001142475TranMinhHoang22082021
- 00020101021BKB0491009260891NguyenVanBa26082021
- 00020101021BKB0882003729910DangVanThanh18072020

4.2.Khoa Hoc May Tinh Bank (KHB)

Khoa Hoc May Tinh Bank (KHB) defines their QR message as a sequence of consecutive information fields as follows:

Prefix numbers**Creation time****Bank name****Account number****Account holder name**

In which:

- **Bank name:** “KHB”
- **Account number:** Is a string of numeric characters with 15 characters in length.
- **Creation time:** Is a string of days, months and years in the format **mmddyyyy**.
Example: 22/08/2021 will be stored as “08222021”

Some examples for KHB's QR code message:

- 0002010102108162021KHB012100002512112DangVu
- 0002010102107312021KHB013900001913490NguyenThinhToan
- 0002010102112202020KHB012100000801332HoangCongAnh

4.3.Hoc Bong Bank (HBB)

Hoc Bong Bank (HBB) defines their QR message as a sequence of consecutive information fields as follows:

Prefix numbers**Account holder name****Account number****Creation time****Bank name**

In which:

- **Bank name:** “HBB”



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

- **Account number:** Is a string of numeric characters with 8 characters in length.
- **Creation time:** Is a string of days, months and years in the format **yymmdd** (yy are the last 2 digits of the year).

Example: 22/08/2021 will be stored as “210822”.

Some examples for HBB's QR code message:

- 00020101021TrinhThiKhanhDuyen29136412200101HBB
- 00020101021HuynhNgocPhu14226317200706HBB
- 00020101021NguyenThanhBinh15883012201130HBB



5. View recipient's information from QR code message

5.1. Function introduction

After scanning the QR code and getting the message, the next thing the tool needs to do is process the information of this raw message. Specifically, this function will do the following:

- Read input as the message of the QR code.
- Identify the format of the message corresponding to the banks mentioned in section 4.
- Process data and print out the information from the message of the QR code.

5.2. Input data

When selecting the View recipient's information from QR code message function, the input data consists of 2 lines. The first line of input data is **Menu_code** and the second line will be a string containing the **Message of the QR code**. The input data has the following format:

1 <Message of the QR code>

5.3. Output data

The output data will be a string that contains many lines. Each line is a piece of information extracted from the raw QR message according to Section 4. Namely, the output data will have the following format:

Bank name: <Bank name> Account number: <Account number> Account holder name: <Account holder name> Creation time: <Creation time>
--

In which:

- **Bank name**, **Account number**, and **Account holder name** will be printed as described in Section 4.
- **Creation time** will be printed with format dd/mm/yyyy (e.g., 19/02/2022).



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

Note that:

- If **Creation time** extracted from a message has only the last 2 digits of the year (refers to Section 4.3), then when printing to the screen, the first 2 digits of the year will be 20. For instance, if **Creation time** in message is “220821”, the program prints to the screen 21/08/2022.
- There should be a blank space right after each colon (“:”)
- All QR messages entered are valid messages. Student should not care about invalid QR messages.



5.4. Some examples for reference

Example 1	
Input	1 00020101021BKB0271001142475TranMinhHoang22082021
Output	Bank name: BKB Account number: 0271001142475 Account holder name: TranMinhHoang Creation time: 22/08/2021

Example 2	
Input	1 0002010102108162021KHB012100002512112DangVu
Output	Bank name: KHB Account number: 012100002512112 Account holder name: DangVu Creation time: 16/08/2021

Example 3	
Input	1 00020101021TrinhThiKhanhDuyen29136412200101HBB
Output	Bank name: HBB Account number: 29136412 Account holder name: TrinhThiKhanhDuyen Creation time: 01/01/2020



6. Convert format of the QR code message

6.1.Function introduction

Although each bank has applied for QR payment, for some reason, a bank's mobile banking application can only recognize its own QR code (which means, cannot read the QR message format of other banks!). Hence, in order to help these applications to recognize other banks' QR messages, our support tool will convert the format of one bank to another. Specifically this function will:

- Specify the bank format you want to convert to (Called Bank B).
- Read the input QR message.
- Identify the format of the message corresponding to the banks mentioned in section 4. (Called Bank A)
- Convert the QR message in format of bank A to bank B.

6.2.Input data

When selecting the Convert format of the QR code message function, the input data consists of 3 lines:

- First line of the input is **Menu_code**.
- The second line is the bank name to convert to (Bank B).
- Third line will be the message of QR code.

The input data has the following format:

2
<Bank to convert to>
<Message of QR code>

In which:

- **Bank to convert to:** Is 1 of 3 following strings: “BKB”, “KHB”, “HBB”



Note that:

- If the **Bank to convert to** does not match 1 of the 3 strings mentioned above, the program prints out "**Invalid bank to convert to**" and terminates the program.

6.3.Output data

The output data is a QR message string in the format of the bank you want to convert to. Students refer to section 4 to understand the data to be output.



6.4. Some examples for reference

Example 4	
Input	2 KHB 00020101021BKB0271001142475TranMinhHoang22082021
Output	0002010102108222021BKB0271001142475TranMinhHoang
Explain	Convert QR code from BKB format to KHB

Example 5	
Input	2 HBB 0002010102107312021KHB013900001913490NguyenThinhToan
Output	00020101021NguyenThinhToan013900001913490210731KHB
Explain	Convert QR code from KHB format to HBB

Example 6	
Input	2 VCB 00020101021HuynhNgocPhu14226317200706HBB
Output	Invalid bank to convert to
Explain	The bank “VCB” is not among the 3 eligible banks



7. Encrypt the message of QR code

7.1.Function introduction

Although the QR code messages of banks have different formats, if we look closely, we can see the rules. Recognizing these rules will be a loophole that exposes the user's personal information. In this function, we will implement the encryption of the messages of the QR code to protect the user's information, namely:

- Specify all encrypting parameters (modes, in short).
- Read the message of QR code
- Print to the screen the encrypted message of QR code.

7.2.Input data

When selecting the Encrypt the message of QR code function, the input data consists of 3 lines:

- First line of the input is **Menu_code**.
- Second line are modes.
- Third line will be the message of QR code.

The input data has the following format:

```
3
<Account holder name mode> <Account number mode> <QR message mode>
< Message of QR code >
```

In which:

- **Account holder name mode**: can be 0, 1 or 2. Refer to section 7.4.
- **Account number mode**: can be 0 or 1. Refer to section 7.5.
- **QR message mode**: can be 0 or 1. Refer to section 7.6.

Note that:

- If there are any input encoding parameters that are not in the above values, the program prints out "**Invalid encrypting parameters**" and terminates the program.



- **QR code encryption** must be done **after** 2 operations of **account holder name encryption** and **account number encryption**.

7.3. Output data

The output data is an encrypted QR message string.

7.4. Account holder name encryption

The **Account holder name mode** is an integer that identifies the encoding for the **Account holder name** information field in the QR code message. This parameter accepts only one of the following 3 values:

- If is 0: No encryption
- If is 1: Account holder name (including uppercase and lowercase characters) is encrypted for characters (column **Char**) to numbers (column **Decimal**) according to the below ASCII Table:

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



Examples:

- TranHoang -> 84114971107211197110103

Original string	T	r	a	n	H	o	a	n	g
Encrypted string	84	114	97	110	72	111	97	110	103

- NguyenVanA -> 78103117121101110869711065

Original string	N	g	u	y	e	n	V	a	n	A
Encrypted string	78	103	117	121	101	110	86	97	110	65

- If is 2: Account holder name is encrypted according to the cipher specification as follows:

- Characters will be replaced by 2-digit numeric strings. Specifically:
 - Characters “A” to “Z” will correspond to “00” to “25”.
 - Characters “a” to “z” will correspond to “26” to “51”.

Examples:

- TranHoang -> 194326390740263932

Original string	T	r	a	n	H	o	a	n	g
Encrypted string	19	43	26	39	07	40	26	39	32

- NguyenVanA -> 13324650303921263900

Original string	N	g	u	y	e	n	V	a	n	A
Encrypted string	13	32	46	50	30	39	21	26	39	00

7.5.Account number encryption

The **account number mode** is an integer specifying the encoding for the **Account number** information field in the QR code message. This parameter accepts only one of the following 2 values:

- If is 0: No encryption
- If is 1: Account number is converted into reverse string compared to the original string

For instance:

- 0271001056475 -> 5746501001720
- 14226317 -> 71362241



7.6. QR message encryption

Again, this **QR message encryption** can only performs **after finishing** the above 2 encryptions.

The **QR message mode** is an integer that identifies the encoding for the entire **QR code message**. This parameter accepts only one of the following 2 values:

- If is 0: No encryption
- If is 1: The message will be encoded using Caesar Encryption. Specifically, the Caesar cipher changes each character from the original text to a new character, positioned an **offset d** in a given alphabet. The Caesar cipher formula can be represented as follows:

$$E(x) = (x + d) \bmod p$$

In which:

- $E_d(x)$: The character after being encoded by shifting d positions
- x : The Character before encoding
- d : offset
- p : Number of letters in **the alphabet**

Regarding the **offset d** , let s be the number of characters of **Account holder name** (If **Account holder name** has been encoded in the above section, the name must be taken **after** the encoding). In this assignment, d is calculated as:

$$d = (11 * s) \bmod p$$

Regarding **the alphabet** used, the upper- and lower-case alphabets are paired in consecutive pairs and finally the digits are as follows:

A	a	B	b	C	c	...	Z	z	0	1	...	9
---	---	---	---	---	---	-----	---	---	---	---	-----	---

For example:

- With $d = 2$: ZzzBaa089Czab -> 011Cbb2AaD1bc

d = 2													
Original string	Z	z	z	B	a	a	0	8	9	C	z	a	b
Encrypted string	0	1	1	C	b	b	2	A	a	D	1	b	c



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

- With $d = 10$: NguyenVanA0002010 -> Slz7js0fsFAAABAaA

d = 10																	
Original string	N	g	u	y	e	n	V	a	n	A	0	0	0	2	0	1	0
Encrypted string	S	l	z	7	j	s	0	f	s	F	A	A	A	B	A	a	A



7.7. Some examples for reference

Example 7	
Input	3 1 0 0 00020101021BKB0271001142475TranMinhHoang22082021
Output	00020101021BKB0271001142475841149711077105110104721119711010322082021
Explain	Encrypt the account holder name in ASCII: TranMinhHoang -> 8411497110771051101047211197110103

Example 8	
Input	3 2 0 0 00020101021BKB0271001142475TranMinhHoang22082021
Output	00020101021BKB02710011424751943263912343933074026393222082021
Explain	Encrypt the account holder name according to the specified cipher: TranMinhHoang -> 19432639123439330740263932

Example 9	
Input	3 0 1 0 00020101021TrinhThiKhanhDuyen29136412200101HBB
Output	00020101021TrinhThiKhanhDuyen21463192200101HBB
Explain	Encrypt account number: 29136412 -> 21463192



Example 10

Input	3 0 0 1 0002010102108162021KHB012100002512112DangVu
Output	444645454654B5A6465MJD456544446956556FcpiXw
Explain	Encrypt entire the QR message with d = 4 (s = 6): 0 -> 4, 2 -> 6, 8 -> B, K -> M, ... (Shift 4 times according to the alphabet in section 7.6)

Example 11

Input	3 1 1 0 00020101021BKB0882003729910DangVanThanh18072020
Output	00020101021BKB019927300288068971101038697110841049711010418072020
Explain	Encrypt account holder name: DangVanThanh -> 689711010386971108410497110104 Encrypt account number: 0882003729910 -> 0199273002880

Example 12

Input	3 1 2 0 00020101021BKB0882003729910DangVanThanh18072020
Output	Invalid encrypting parameters
Explain	Account number mode does not have the value 2



Example 13

Input	3 0 1 1 00020101021NguyenThanhBinh15883012201130HBB
Output	pppppQpQpQc2KOZDi4VD4v6D4qQpRttSQqpQQRp3vv
Explain	Encrypt account number: 15883012 -> 21038851 QR message after performing 2 encrypting steps (account holder name and account number): 00020101021NguyenThanhBinh21038851201130HBB Encrypt entire the above QR message with d = 41 (s = 15) : 0 -> p, N -> c, a -> V, B -> v, ... (Shift 41 times according to the alphabet in section 7.6)

Example 14

Input	3 2 1 1 00020101021BKB0882003729910DangVanThanh18072020
Output	DDDEDdDdDEdJSJDdhhEgeDDEHHDDeEGeHeEEdeEGehdheeEGeheedHDgEDED
Explain	Encrypt the account holder name according to the specified cipher: DangVanThanh -> 689711010386971108410497110104 Encrypt account number: 0882003729910 -> 0199273002880 QR message after performing 2 encrypting steps (account holder name and account number): 00020101021BKB019927300288003263932212639193326393318072020 Encrypt entire the above QR message with d = 16 (s = 24) : 0 -> D, B -> J, 9 -> h, 1 -> d, ... (Shift 16 times according to the alphabet in section 7.6)



8. Submission

Students download file *NMLT assignment - English.zip* from the e-Learning site. When extracting this file, you will get the following files:

qrTool.c	Initial source code
[NMLT]QR-Assignment - English.pdf	Assignment specifications

The qrTool.c file is the initial source code. Students are encouraged to use this source code for further implementation.

When submitting, students submit their work on the subject's e-Learning site. Students fill in the code for the assignment like other exercises. Students are provided with 4 places to submit:

- Part 1 - View QR information (**Assignment – View QR information**): Students submit their work and are graded on **30 testcases** related to the View QR information section.
- Part 2 - Convert QR format (**Assignment – Convert QR format**): Students submit their work and are graded on **30 testcases** related to the Convert QR format section.
- Part 3 - Encrypt QR message (**Assignment – Encrypt QR message**): Students submit their work and are graded on **20 testcases** related to the Encrypt QR message section.
- Part 4 – Mixed tests (**Assignment – Mixed tests**): This part includes **20 testcases** that test all 3 parts above, including some difficult testcases. Students need to achieve a minimum of **7 points** in **all 3 sections** above to be able to submit work to the **Mixed tests**. Students who do not submit their work to the **Mixed tests** will be counted as 0 points for only this part.

In each of the above parts, students have a maximum of **10 attempts**. For each attempt, students have **10 minutes** to check and submit the code. Only the first "Check" is scored, the following times will not be scored. The test results are displayed only after you press the "Complete" button. The highest score among attempts will be taken as the score for that part.



Assignment score is calculated according to the following formula:

$$\text{Assignment score} = \text{Part 1 score} * 0.3 + \text{Part 2 score} * 0.3 + \text{Part 3 score} * 0.2 + \text{Part 4 score} * 0.2$$

For example, student A gets 10 points in Part 1, 7 points in Part 2 and 8 points in Part 3.

Student A is eligible to submit Part 4, where A's result is 6 points. Hence:

$$\text{A's Assignment score} = 10 * 0.3 + 7 * 0.3 + 8 * 0.2 + 6 * 0.2 = 7.9$$

**Students must test their program on MinGW
and Assignment – Test place before submitting.**

8.1.Submission deadline

Deadline for submission is **11:55 p.m. Sunday, November 7, 2021**. Students submit work on the e-Learning site of the subject. If a student submits assignment late, the maximum score for assignment will be **0 points**. We DO NOT accept submissions sent via email or any other form.

8.2.Handling fraud

Assignment must be done BY YOURSELF. Students will be considered fraudulent if:

- There is an unusual similarity between the source code of the submissions. In this case, ALL submissions are considered fraudulent. Therefore, students must protect the source code of their assignments.
- Students do not understand the source code written by themselves, except for the parts of the code provided in the initialization program. Students can consult from any source, but make sure they understand the meaning of all the lines they write. In case of not understanding the source code where they refer, students are especially warned NOT to use this source code; instead use what has been learned to write programs.
- Mistakenly submit another student's assignment on your personal account.

In the case of cheating, students will get a 0 for the entire subject (not just the assignment).

WE DO NOT ACCEPT ANY INTERPRETATION AND NO EXCEPTION!

After assignment has been submitted, a number of students will be called for random interviews to prove that the assignment has been done by themselves.

-FIN-



9. Changelog

v1.1:

- Add Changelog
- Add Test place