

1. Design and develop an assembly language to search a key element "x" in a list of 'n' 16-bit numbers. Adopt Binary Search algorithm in your program for searching.

.model small

init ds macro

 mov ax, @data

 mov ds, ax

endm

printf macro msg

 lea dx, msg

 mov ah, 9

 int 91h

endm

putchar macro char

 mov dl, char

 mov ah, 2

 int 91h

endm

exit macro

 mov ah, 4ch

 int 21h

endm

.data

array dw 1122h, 2345h, 3333h, 4455h, 6666h
 len dw (\$-array)/2

search equ 2345h

foundmsg db 'Element found at position: \$'
 position db 0

notfoundmsg db 'Element not found \$'

.code

initds .

mov bx, 1

mov dx, len

mov cx, search

again:

cmp bx, dx

ja failure

mov ax, bx

add ax, dx

shl ax, 1

mov si, ax

dec si

add si, si

Teacher's Signature : _____

Expt. No.

Date

Page No. 03

cmp $\cdot(\text{cx}, \text{array}[\text{si}])$
jae bigger

dec ax
mov dx, ax
jmp again

bigger:

je success
inc ax
mov bx, ax
jmp again

success:

add al, 30h
mov position, al
printf "found msg"
putchar position
exit

failure:

printf "not found msg"
exit

end

OUTPUT:-

onasm P1.asm;

link P1.obj;

P1

Element found at position : 2

Q. Design and Develop an assembly program to sort a given set of 'm' 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.

.model small

.initds ormacao

· mov ax, @data

· mov ds, ax

Endam

.data

array dw 80h, 70h, 40h, 10h, 50h

Count dw (\$ - array) / 2

.code

initds

· mov dx, Count

· dec dx

outerloop:

· mov cx, dx

· lea si, array

innerloop:

· mov bx, [si]

Teacher's Signature: _____

comp $\cdot ax, [sita]$

jl swap

xchg $[sita], ax$

mov $[si], ax$

swap:

add $si, 02$

loop innerloop

dec dx

jnz outerloop

int 3

align 16

end

OUTPUT:

.o main pg.asm;

link pg.obj;

CV pg.exe

press F5 @ g -> (g means go and execute)

d d8:0 -> (d means dump, d8 means data segment)

rd d8:0

3A07:0000 10 00 90 00 40 00 50 00 70 00 05 00 00 00
3A07:0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Note:-

d. d8:0 means dump the data segment from 0th location.

3. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.

.model small

```
initds macro  
    mov ax, @data  
    mov ds, ax
```

endm

```
inites macro  
    mov es, ax
```

endm

```
printf macro msg  
    //lea dx, msg  
    mov ah, 9  
    int 91h
```

endm

```
getchar macro  
    mov ah, 1  
    int 91h
```

endm

```
exit macro  
    mov ah, 4ch  
    int 91h
```

endm

Teacher's Signature : _____

. data

original db 30 dup(?)

reverse db 30 dup(?)

ask db 10,13, "String please: \$"

palindromemsg db 10,13, "Palindrome \$"

notpalindromemsg db 10,13, "Not palindrome \$"

. code

init ds

init es

lea si, original

lea di, reverse

printf ask

mov cx, 00

take input:

getchar

cmp al, 13

je done

mov [si], al

inc cx

inc si

jmp takeinput

done: dec si

Teacher's Signature : _____

reverseing task:

mov al,[si]

mov [di],al

inc di

dec si

jmp reverseing task

lea si,original

lea di,reverse

cld

repeat

je palion

paintl : notpalindromesng

exit

palion : paintl : palindromesng

exit

end

OUTPUT 1:-

P3.exe

String : please : MADAM

Palindrome

OUTPUT 2:-

P3.exe

String : please : COLLEGE

Not Palindrome

4. Develop an assembly language program to compute product of two non-negative integers i.e. Assume that 'm' and 'n' are non-negative integers.

.model small

init ds small

mov ax, @data

mov ds, ax

endm

putchar macro char

mov dl, char

mov ah, 2

int 91h

endm

exit macro

mov ah, 4ch

int 91h

endm

, data

m db 6

n db 3

answer db 0

.code

initds

mov al, on

mov bl, n

call mca

mov al, answer

add ; adjust after multiplication

add ax, 3030h

mov bx, ax

pusha .bh

pusha bl

exit

main proc

cmp bl, 0

jne go1

; $m_{c_0} = 1$

add answer, 1

ret

go1 : cmp bl, 0

jne go2

; $m_{c_n} = 1$

add answer, 1

ret

go2 : cmp bl, 1

jne go3

; $m_{c_t} = m$

add answer, 1

ret

Teacher's Signature : _____

go3: mov dec al

comp bl,al

jne go4

$j \text{ c}_{m-1} = m$

inc al

add answer,al

ret

go4: push ax

push bx

call mca

$j \text{ c}_{m-1}$

pop bx

pop ax

- dec bx

push ax

push bx

call mca

pop bx

pop ax

ret

mca endp

end

OUTPUT:

$${}^n C_r = \frac{n!}{r!(n-r)!}$$

maxm p4.asm;

link p4.obj;

p4

20

5. Design and Develop an assembly language program to read the current time and date from the system and display it in the standard format on the screen.

model small

in1ds imac386

mov ax, @data

mov ds, ax

endm

printf macro msg

lea dx, msg

mov ah, 9

int 91h

endm

putchar macro char

mov dl, char

mov ah, 9

int 91h

endm

accetime macro

mov ah, 0ch

int 91h

endm

accessdate mado

mov ah, 2ah
int 21h

endm

display : mado value

mov al, value

add

add ax, 3030h

mov bx, ax

putchar bh

putchar bl

endm

time mado

paintl : timemsg

access time

display ch

putchar :

display dl

endm

date mado

paintl : datemsg

access date

display dl

putchar :

display dh

endm

Teacher's Signature : _____

Date

Expt. No.

Page No. 14

. data

timemsg db 10,13,"Current time is %d : "

datemsg db 10,13,"Current date is %d "

. code

initds

time

date

exit

end

Output:

PS. EXE

Current time is 10:37

Current date is 14/03

6 To write and Simulate ARM assembly language program for data transfer, arithmetic and logical operations.
(Demonstrate with the help of suitable program)

AREA .PRG6, CODE, READONLY

ENTRY ; the entry point where the code starts

LDR R0,=5 ; data transfer - R0=5
LDR R1,=3 R1=3

ADD R2,R0,R1 ; arithmetic ADD - R2=8 (5+3)
SUB R3,R0,R1 ; SUB R3=2 .(5-3)
MUL R4,R0,R1 ; MUL R4=F (5*3=15=F)

AND R5,R0,R1 ; logical AND - R5=1 (5&3)
ORR R6,R0,R1 ; OR R6=7 (5|13)
EOR R7,R0,R1 ; XOR R7=6 (5^3)

Output:

Press F7, then press Ctrl+F5 (Start debug session) and
Keep pressing F11. You will notice the following.

Register	Value
R0	0x00000005
R1	0x00000003
R2	0x00000008
R3	0x00000002
R4	0x0000000F
R5	0x00000001
R6	0x00000007
R7	0x00000006

7 To write and simulate C program for ARM microprocessor
using KEIL (Demonstrate with the help of a suitable program)

#include <LPC914X.h>

```
int main()
{
    int a,b,c,d,e,f;
    a=5;
    b=3;
    c=a+b;
    d=a-b;
    e=a*b;
    f=a/b;
    return 0;
}
```

OUTPUT:-

press F7, then press ShiftF5 (Start debug session) and F11. You'll notice the following

Name	Location/Value	Type
main	0x00000104	int f()
a	0x00000005	auto-int
b	0x00000003	auto-int
c	0x00000008	auto-int
d	0x00000002	auto-int
e	0x0000000F	auto-int
f	0x00000001	auto-int