**An**

**Assignment**

**On**

**“UVA Problem Solutions Using C and Assembly Language”**

**Assembly Language Laboratory**

**[ CSE 2208]**

**Submitted To: -**

Sarfaraz Newaz

Lecturer

Computer Science and Engineering Discipline,

Khulna University.

**Submitted By: -**

Md Rakibul Hasan Molla

Student ID: 1602006

Year: 3rd Term: II

Computer Science and Engineering Discipline,

Khulna University.

Date: 22 October 2018



Computer Science and Engineering Discipline,

Khulna University

**Index**

1. Hashmat the Brave Warrior (10055) 3
2. Ugly Number (136) 7
3. A Benevolent Josephus (1176) 10
4. 1727
5. 11984
6. 12478
7. 12704
8. 299
9. 694
10. 10055
11. 10071
12. 10079
13. 10346
14. 10499
15. 10696
16. 10783
17. 10812
18. 10929
19. 10970
20. 11172
21. 11219
22. 11479
23. 11727
24. 11799
25. 11805
26. 11854
27. 11875
28. 11879
29. 12372
30. 12468
31. 12577
32. 12646
33. 12708

**Problem name:** Hashmat the Brave Warrior (10055)

**C Solution:**

#include<stdio.h>

int main() {

long long a, b;

while(scanf("%lld %lld", &a, &b) == 2) {

if(a > b) printf("%lld\n", a-b);

else printf("%lld\n", b-a);

}

return 0;

}

**Assembly Solution:**

|  |  |
| --- | --- |
| **.MODEL SMALL**  **.STACK 100H**  **.CODE**  **MAIN PROC**      **START:**  **CALL SCAN**    **MOV BH, DL ;move first value in bh**  **CALL SCAN**  **MOV BL, DL ;move second value in bl**  **SUB BH, BL ;SUBTRACT THE INPUT VALUES**  **MOV AX, 0 ;CLEAR AX**  **MOV AL, BH ;MOVE SUBTRACTION TO AL FOR PRINT PROC**  **CMP AL, 0**  **JG P ;IF SUBTRACTION IS NOT GREATER THAN ZERO**  **NEG AL ;NEGATE SUBTRACTION**        **P:**    **CALL PRINT**  **MOV AH, 2**  **MOV DL, 10 ;PRINT NEW LINE**  **INT 21H**  **MOV DL, 13 ;PRINT CARRIGE RETURN**  **INT 21H**  **JMP START ;TAKE INPUT AGAIN**  **RET**  **MAIN ENDP**  **;a procedure that read an integer value**  **;of one or more digit**  **;input is terminated by both space and new line**  **;the inputed integer will be present in dl** | **SCAN PROC**  **MOV DX, 0**  **INPUT:**  **MOV AH, 1**  **INT 21h**  **CMP AL, ' '**  **JE END**  **CMP AL, 13**  **JE END**  **PUSH AX**  **MOV AL, 10**  **MUL DL**  **MOV DL, AL**  **POP AX**  **SUB AL, '0'**  **ADD DL, AL**    **JMP INPUT**  **END:**  **RET**  **SCAN ENDP**    **;a procedure that prints an integer**  **;consisting of one or more digits**  **;the integer must be present in ax**  **PRINT PROC**    **XOR CX, CX**  **LOOP1:**  **CWD**  **MOV BX, 10**  **IDIV BX**  **PUSH DX**  **INC CX**  **CMP AX, 0**  **JG LOOP1**    **LOOP2:**  **POP DX**  **ADD DX, '0'**  **MOV AH, 2**  **INT 21h**  **LOOP LOOP2**  **RET**  **PRINT ENDP** |

**Problem name:** Back to High School Physics -10071

**C Solution:**

#include<stdio.h>

int main() {

long long v, t;

while(scanf("%lld %lld", &v, &t) == 2)

printf("%lld\n", 2\*v\*t);

return 0;

}

**Problem name:** Pizza Cutting -10079

**C Solution:**

#include<stdio.h>

int main() {

long long int k;

while(scanf("%lld", &k) == 1 && k >= 0) {

printf("%lld\n", k\*(k+1)/2+1);

}

return 0;

}

**Problem name:** The Land Of Justice - 10499

**C Solution:**

#include<stdio.h>

using namespace std;

int main(){

long int n,r;

while(scanf("%ld",&n)==1){

if(n<0) break;

if(n==1) r=0;

else r=n\*25;

printf("%ld%%\n",r);

}

return 0;

}

//**UVa Problem**-10970(Big Chocolate)

//Accepted

//Running time: 0.260 sec

#include<iostream>

using namespace std;

int main(){

long m,n;

while(cin>>m>>n){

cout<<(m\*n)-1<<endl;

}

return 0;

}

**11172 – Relational Operator**

#include<stdio.h>

int main() {

int T, x, y;

scanf("%d", &T);

while(T--) {

scanf("%d %d", &x, &y);

if(x < y) puts("<");

else if(x > y) puts(">");

else puts("=");

}

return 0;

}

**11799- Horror Dash**

#include <stdio.h>

int main() {

int t, n, Case = 0;

scanf("%d", &t);

while(t--) {

scanf("%d", &n);

int max = 0, c;

while(n--) {

scanf("%d", &c);

if(max < c)

max = c;

}

printf("Case %d: %d\n", ++Case, max);

}

return 0;

}

**11805**

#include <stdio.h>

int main() {

int t, test = 0, N, K, P;

scanf("%d", &t);

while(t--) {

scanf("%d %d %d", &N, &K, &P);

printf("Case %d: %d\n", ++test, (K+P-1)%N+1);

}

return 0;

}