

3. A university is receiving applications from students to register for language courses offered in six different languages. Each language has limited seats and time of submission of request is to be used for making a decision. The following are the fields provided in the application:
- Register Number, Name, Time and date of submission, Preference of language {Option1, Option2, Option3}
 - Devise a solution that would automate the process of allocating the languages based on the order of preference given by students. Use appropriate data structures to process the data. Include provision for swapping of the language that has been allocated for a student on mutual consent.

Gale-Shapley Algorithm:

The idea is to iterate through all free languages while there is any free language available. Every free language goes to all students in their preference list according to the order. For every student language goes to check if the student is free, if yes, then that student is taken by the language.

If in case the student is not free then the student's preference list is checked accordingly if language comes first in preference list then the student is taken else not free. Hence a combination of student and language done once can be broken later and can be made into a new one.

Time Complexity- $O(n^2)$

Input & output-

Input is a 2D matrix of size $(2*N)*N$ where N is the number of students and languages as per the question $N=6$.

Rows from $0-N-1$ represent preference lists of languages and rows from N to $2*N-1$ represents the preference list of students.

So languages are numbered from 0 to $N-1$ and students are numbered from N to $2*N-1$.

The output is the list of pairs of student Reg. No. and languages given.

```
//C program for stable matching and allocation of language
#include<stdio.h>
#include<string.h>
#include<stdbool.h>

//Number of students and languages
#define N 6

//this function returns true if student 'istud' prefers language
'ilang1' over 'ilang'
int sPreferL1OverL(int iprefer[2*N][N],int istud,int ilang, int
ilang1)
{
    int i;
```

```

    //Check if istud prefers ilang over his current engagement
    ilang1
    for (i=0; i<N; i++)
    {

        /*If ilang1 comes before ilang in the preference
        list of istud, then istud prefers his
        current engagement, don't do anything*/
        if (iprefer[istud][i]==ilang1)
            return true;

        /*if ilang comes before ilang1 in istud list,
        then free his current engagement and engage him with ilang*/
        if (iprefer[istud][i]==ilang)
            return false;
    }
}

/*Prints stable matching for N languages and N students.
languages are numbered as 0 to N-1. Students are numbered as N
to 2N-1.*/
void stablePair(int iprefer[2*N][N])
{
    /*Stores language chosen by a student. This is our output
    array that stores information. The value of istudLang[i]
    indicates the language assigned to student N+i. Note that
    the student numbers between N and 2*N-1. The value -1
    indicates that (N+i)th student is free*/
    int istudLang[N];

    /*An array to store availability of languages. If ilangFree[i]
    is
    false, the language 'i' is free, otherwise engaged.*/
    int ilangFree[N];

    //Initialize all languages and students as free
    //memset() is used to fill a block of memory with a particular
    value.
    memset(istudLang, -1, sizeof(istudLang));
    memset(ilangFree, false, sizeof(ilangFree));
    int freeCount=N;

    //while there are free languages
    while (freeCount>0)
    {
        //Pick the first free language (we could pick any)
        int ilang;

```

```

        for(ilang=0;ilang<N;ilang++)
            if(ilangFree[ilang]==false)
                break;

        /*One by one go to all students according to languages
preferences.
        Here ilang is the picked free language*/
        int i;
        for(i=0;i<N && ilangFree[ilang]==false;i++)
        {
            int istud = iprefer[ilang][i];

            /*The student of preference is free, istud and ilang
become engaged (Note that the engagement can be changed
later).
            So we can say they are engaged not permanently
together.*/
            if(istudLang[istud-N]==-1)
            {
                istudLang[istud-N]=ilang;
                ilangFree[ilang]=true;
                freeCount--;
            }
            else //If istud is not free
            {
                //Find current engagement of istud
                int ilang1=istudLang[istud-N];

                /*If istud prefers ilang over his current engagement
ilang1,
                then break the engagement between istud and ilang1
and
                enagage ilang with istud.*/

                if(sPreferL1OverL(iprefer,istud,ilang,ilang1)==false)
                {
                    istudLang[istud-N]=ilang;
                    ilangFree[ilang]=true;
                    ilangFree[ilang1]=false;
                }
            } //End of Else
        } //End of the for loop that goes to all students in
languages list
    } //End of main while loop

//Print the solution

```

```

printf("*****LANGUAGES OFFERED AND THEIR NUMERIC
CODES*****");
printf("\n\t\t\t\tSpanish-0\n\t\t\t\tChinese-1\n\t\t\t\tGerman-
2\n\t\t\t\tJapanese-3\n\t\t\t\tKorean-4\n\t\t\t\tFrench-5\n");
printf("\n");
printf("Reg.No.   Name   Time of submission   Date of submission
Preference of Language\n");
printf("   6       Tina           20:00           01/2/21           {5,
4, 3, 2, 1, 0}\n");
printf("   7       Rina           08:28           09/3/21           {4,
3, 2, 1, 0, 5}\n");
printf("   8       Mina           14:15           07/4/21           {3,
2, 1, 0, 5, 4}\n");
printf("   9       Lina           03:30           02/5/21           {2,
1, 0, 5, 4, 3}\n");
printf("  10       Jina           21:00           10/6/21           {1,
0, 5, 4, 3, 2}\n");
printf("  11       Bina           09:34           29/7/21           {0,
5, 4, 3, 2, 1}\n");
printf("\n");
printf("*****LANGUAGE GIVEN*****\n");
printf("Student Reg.No.       Language given\n");
int i;
for(i=0;i<N;i++)
printf("          %d              %d\n",i+N,istudLang[i]);

}

```

//Drivers program to test above functions

```

int main()
{
    int iprefer[2*N][N]={ {11,10,9,8,7,6},
        {10,9,8,7,6,11},
        {9,8,7,6,11,10},
        {8,7,6,11,10,9},
        {7,6,11,10,9,8},
        {6,11,10,9,8,7}, // Student reg. No.
        {0,1,2,3,4,5},
        {0,1,2,3,4,5},
        {0,1,2,3,4,5},
        {0,1,2,3,4,5},
        {0,1,2,3,4,5},
        {0,1,2,3,4,5}, //language codes
    };
    stablePair(iprefer);
    return 0;
}

```

```
"E:\c\Stable matching.exe"
*****LANGUAGES OFFERED AND THEIR NUMERIC CODES*****
Spanish-0
Chinese-1
German-2
Japanese-3
Korean-4
French-5

Reg.No.  Name  Time of submission  Date of submission  Preference of Language
6      Tina   20:00              01/2/21             {5, 4, 3, 2, 1, 0}
7      Rina   08:28              09/3/21             {4, 3, 2, 1, 0, 5}
8      Mina    14:15              07/4/21             {3, 2, 1, 0, 5, 4}
9      Lina    03:30              02/5/21             {2, 1, 0, 5, 4, 3}
10     Jina    21:00              10/6/21             {1, 0, 5, 4, 3, 2}
11     Bina    09:34              29/7/21             {0, 5, 4, 3, 2, 1}

*****LANGUAGE GIVEN*****
Student Reg.No.    Language given
6          5
7          4
8          3
9          2
10         1
11         0

Process returned 0 (0x0)   execution time : 4.117 s
Press any key to continue.
```