Institute of Computer Technology

B. Tech Computer Science and Engineering

Subject: ESFP-II (2CSE203)

**PRACTICAL-1**

**AIM: - To learn about Dynamic Memory allocation.**

***Exercise: -***

**1. I wants to find the largest value from the list and all list elements are assigned memory at runtime not compile time.**

**Test Data:**

**-------------------------------------------------------------------------**

**Input total number of elements (1 to 100): 5**

**Number 1: 5**

**Number 2: 7**

**Number 3: 2**

**Number 4: 9**

**Number 5: 8**

**The Largest element is: 9.00**

***CODE:***

#include <iostream>

#include <conio.h>

#include <cstdlib>

using namespace std;

int main()

{

    int i,n;

    int \*num;

    printf("Input total number of elements(1 to 100):");

    scanf("%d",&n);

    num=(int\*) calloc(n, sizeof(int));

    if(num==NULL) {

        printf("No memory is allocated.\n");

        exit(0);

    }

    printf("\n");

    for (i = 0; i < n; ++i)

    {

        printf("Number %d: ",i+1);

        scanf("%d",num+i);

    }

    for (i = 1; i < n; ++i)

    {

        if(\*num<\*(num+i)) {

            \*num=\*(num+i);

        }

    }

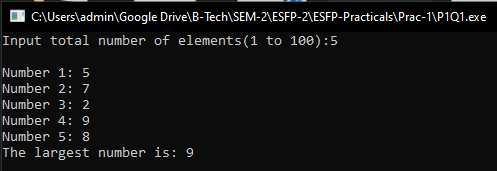
    printf("The largest number is: %d\n",\*num);

getch();

    return 0;

}

***OUTPUT:***

****

**2. Developer wants to take input text and print that text using reallocated memory & memory which can be released by compiler.**

***CODE:***

#include <iostream>

#include <cstdlib>

#include <string>

#include <conio.h>

using namespace std;

int main() {

    int n;

    char \*name;

    cout<<"Enter a size of string"<<endl;

    cin>>n;

    name=(char\*) malloc(n\*sizeof(char));

    cout<<"Enter a string=";

    fflush(stdin); //buffered memory is cleared

    gets(name);

    cout<<"You entered= "<<name<<endl;

    cout<<"Enter resize of string length: ";

    cin>>n;

    name=(char\*)realloc(name, sizeof(char));

    cout<<"Enter a string: ";

    fflush(stdin); //buffered memory clear

    gets(name);

    cout<<"\nYou entered="<<name<<endl;

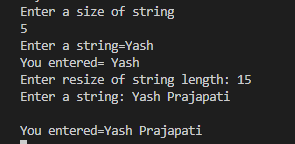
    free(name);

    getch();

    return 0;

}

***OUTPUT:***

******

**3. How can we create n number of strings with m length using runtime memory?**

***CODE:***

#include <iostream>

#include <cstring>

#include <cstdlib>

#include <conio.h>

using namespace std;

struct student

{

char name[10];

int m[3];

int total;

char result[5];

}\*p,\*s;

int main()

{

int i,j,l,n;

cout<<"Enter the no. of students : ";

cin>>n;

p=(struct student\*)malloc(n\*sizeof(struct student));

s=p;

for(i=0;i<n;i++)

{

cout<<"Enter a name : ";

scanf("%s",&p->name);

p-> total=0;l=0;

for(j=0;j<3;j++)

{

one:

cout<<"Enter Marks of "<<j+1<<" Subject : ";

scanf("%d",&p->m[j]);

if((p->m[j])>100)

{

cout<<"Wrong Value Entered";

goto one;

}

p->total+=p->m[j];

if(p->m[j]<40)

l=1;

}

if(l==0)

strcpy(p->result,"PASS");

else

strcpy(p->result,"FAIL");

p++;

}

for(i=0;i<n;i++)

{

printf("\n%s\t%s",s->name,s->result);

s++;

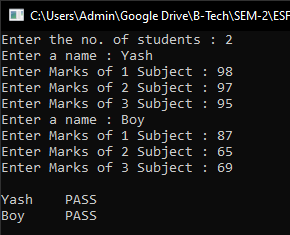
}

getch();

return 0;

}

***OUTPUT:***

****

**Post Practical Work:**

1. How will you free the memory allocated by the following program?

#include<stdio.h>

#include<stdlib.h>

#define MAXROW 3

#define MAXCOL 4

int main()

{

int \*\*p, i, j;

p = (int \*\*) malloc(MAXROW \* sizeof(int\*));

return 0;

}

A. memfree(int p);

B. dealloc(p);

C. malloc(p, 0);

***D. free(p);***

2. Assume integer is 2 bytes wide. How many bytes will be allocated for the following code?

#include<stdio.h>

#include<stdlib.h>

#define MAXROW 3

#define MAXCOL 4

int main()

{

int (\*p)[MAXCOL];

p = (int (\*) [MAXCOL])malloc(MAXROW \*sizeof(\*p));

return 0;

}

A. 56 bytes

B. 128 bytes

***C. 24 bytes***

D. 12 bytes

3. How many bytes of memory will the following code reserve?

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*p;

p = (int \*)malloc(256 \* 256);

if(p == NULL)

printf("Allocation failed");

return 0;

}

A. 65536

***B. Allocation failed***

C. Error

D. No output