```
for(i=size2 ; i<newsize ; i++){
    arr2[i]=i*4 ;
    printf("%d ", arr2[i]);
}
printf("\n");

free(arr2);

return 0;
}
```

<u>output :</u>

```
using malloc :
0 2 4 6 8

using calloc :
0 3 6 9 12 15 18

using realloc :
28 32 36.
```

[3]

```
// Stack implementation

#include <stdio.h>
#include <stdlib.h>

struct stack{
    int *arr ;
    int top,i ;
    int capacity;
};
```

```
void intialize(struck stack * stack, int capacity)
{
  stack -> arr = (int *)malloc(capacity * sizeof
                                        (int));
  stack -> top = -1 ;
  stack -> capacity = capacity;

int isEMpty (struct stack * stack) {
    rturn( stack -> top == -1);
 }

int isFull (Struct stack * stack) {
      return (stack -> top = stock -> capacity - 1);
}

// Function to push an element onto the steck
void push ( struct stack * stack, int value ) {
   if ( isFull(stack) ) {
        printf(" stack overflow, cannot push %d \n",
                                       value);
   } else {
        printf(" %d pushed to the stack, \n", value)
                                                  ;
   }
}

// Function to pop an element from stock
void pop (struct stack * stack) {
  if (is Empty (stack)) {
       printf(" stack underflow, cannot pop
              from an empty stack. \n");
       retun -1;
  }
```

```
else {
    int popedvalue = stack -> arr[stack->top--];
    printf("%d popped from the stack.\n",
                    poppedvalue);
    return popped value;
  }
}

void display(struct stack * stack) {
    if (isEmpty (stack)) {
         printf("Stack is empty.\n");
    } else {
         printf("Elements of the stack:\n");
         for (i=0; i<= stack->top; ++i){
              printf("%d", stack->arr[i]);
         }
         printf("\n");
    }
}

int main() {
 Print f(" Raghavendra.N \n");
 Printf("USN 1BM22CS213 \n");

 int capacity;
printf("Enter the capacity of the stack:");
scanf("%d", &capacity);

struct stack mystack;
initialize(&mystack, capacity);

int numElements;
Printf("Enter the number of elements
              to pus onto stack:");
    scanf("%d", &numElements);

for (i=0; i<numElements; ++i) {
    int element;
    printf("Enter element %d: ", i+1);
    scanf("%d", &element);
     push(&mystack, element);

    display(&mystack);
}

pop(&mystack);

display(&mystack);

free(mystack.arr);

return 0;

}
```

Sample input output;

Raghavendra. N
USN 1BM22CS213

```
Enter the capacity of the stack : 5
Enter the number of elements to push onto
     the stack : 4
Enter element 1 : 10
10 Pushed to the stack.
Elements of the stack :
10
```

Error Next week update
[illegible] 21/12/2023

```
Enter Element : 2 : 20
20 Pushed to the stack.
Elements of the stack.
10 20
Enter element 3 : 30
30 Pushed to the stack
Elements of the stack
10   20   30
Enter element : 4 : 40
40 Pushed to the stack
Elements of the stack
10   20   30  40
Enter POP Element : 10
10 POPed from the stack
Elements of the stack
10  20   30
```

[illegible]
21/12/23