```
//-----------------------------          ---------------------------
int main()
{
    int** a;                  билет 1.1
    int i, j, k,n;
    scanf_s("%d", &n);
    a = (int**)malloc(n * (sizeof(int*)));
    for (i = 0; i < n; i++)
        a[i] = (int*)malloc(n * (sizeof(int)));
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            a[i][j] = rand() % 20 + 10;
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("\n");
    i = j = 0;
    while (i < n)
    {
        while (j < n-1-i)
        {
            int c;
            c = a[i][j];
            a[i][j] = a[n - 1 - j][n - 1 - i];
            a[n - 1 - j][n - 1 - i] = c;
            j++;
        }
        i++;
        j = 0;
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}
//----------------------------- билет 2,2--------------------------
void func(char** a, int n);
int main()
{
    char** a;
    int i, j, n;
    scanf_s("%d", &n);
    a = (char**)malloc(n * sizeof(char**));
    for (i = 0; i < n; i++)
```

```
57          a[i] = (char*)malloc(1 * sizeof(char*));
58      rewind(stdin);
59      j = 0;
60      for (i = 0; i < n; i++)
61      {
62          rewind(stdin);
63          while ((a[i][j++] = getchar()) != '\n')
64          {
65              a[i] = (char*)realloc(a[i], (j + 1) * sizeof(char*));
66          }
67          a[i][j - 1] = '\0';
68          j = 0;
69      }
70      func(a, n);
71  }
72  void func(char** a, int n)
73  {
74      int  i=0,in=0,ik=0,length=0,lengthmax=0,k,inmax,ikmax,j=0;
75      while (n--)
76      {
77          while (a[i][j] == ' ')
78          {
79              in = ++j;
80          }
81          while (a[i][j] != ' ' && a[i][j])
82          {
83              ik = j++;
84          }
85          length = ik - in + 1;
86          if (length > lengthmax)
87          {
88              lengthmax = length;
89              inmax = in;
90              ikmax = ik;
91              k = i;
92          }
93          i++;
94          j = 0;
95      }
96      for (i = inmax; i <= ikmax; i++)
97          printf("%c", a[k][i]);
98  }
99  //----------------------------билет 2.1----------------------------
100 void func(char* a);
101 int main()
102 {
103     char* a;
104     int i = 0, k = 1;
105     a = (char*)malloc(1);
106     while ((a[i++] = getchar()) != '\n')
107         a = (char*)realloc(a, i + 1);
108     a[i - 1] = '\0';
109     func(a);
110 }
111 void func(char* a)
112 {
```

```
    int i=0, s=0, sum=0,k=0;
    while (a[i])
    {
        while (a[i] != ' ' && a[i])
        {
            s+= a[i] - '0';
            s *= 10;
            i++;
        }
        if (a[i] == ' ')
            i++;
        s /= 10;
        sum += s;
        s = 0;
    }
    printf("%d", sum);
}
//------------------------------Билет 2.2------------------------------
void func(int** a, int** b, int n, int n1, int m, int m1,int**c);
int main()
{
    int** a, ** b, ** c;
    int n, m, n1, m1,i,j;
    scanf_s("%d", &n);
    scanf_s("%d", &m);
    scanf_s("%d", &n1);
    scanf_s("%d", &m1);
    a = (int**)malloc(n * sizeof(int**));
    b = (int**)malloc(n1 * sizeof(int**));
    c = (int**)malloc(n * sizeof(int**));
    for (i = 0; i < n; i++)
    {
        a[i] = (int*)malloc(m * sizeof(int*));
        c[i] = (int*)malloc(m1 * sizeof(int*));
    }
    for(i=0;i<n1;i++)
    b[i] = (int*)malloc(m1 * sizeof(int*));
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            a[i][j] = rand() % 10 + 1;
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("\n");
    for (i = 0; i < n1; i++)
    {
        for (j = 0; j < m1; j++)
        {
            b[i][j] = rand() % 10 + 1;
            printf("%d ", b[i][j]);
        }
        printf("\n");
```

```
169     }
170     for (i = 0; i < n; i++)
171     {
172         for (j = 0; j < m1; j++)
173         {
174             c[i][j] = 0;
175         }
176     }
177     func(a, b, n, n1, m, m1,c);
178 }
179 void func(int** a, int** b, int n, int n1, int m, int m1,int**c)
180 {
181     int i=0, j=0, k, s,i1=0,j1=0,i2=0,j2=0;
182     do
183     {
184         do
185         {
186             do
187             {
188                 c[i][j] += a[i1][j1] * b[i2][j2];
189                 j1++;
190                 i2++;
191             } while (j1<m);
192             i2 = 0;
193             j1 = 0;
194             j2++;
195             j++;
196         } while (j<m1);
197         i2 = j2=j1= 0;
198         i1++;
199         i++;
200         j = 0;
201     } while (i<n);
202     for (i = 0; i < n; i++)
203     {
204         for (j = 0; j < m1; j++)
205         {
206             printf("%d ", c[i][j]);
207         }
208         printf("\n");
209     }
210 }
211 //----------------------------билет 3.1----------------------------
212 void func(int* a, int* b, int n);
213 int main()
214 {
215     int* a, * b,n;
216     scanf_s("%d", &n);
217     a = (int*)malloc(n * sizeof(int*));
218     int i,j=0;
219     int k;
220     for (i = 0; i < n; i++)
221     {
222         scanf_s("%d", &a[i]);
223         j = i;
224         k = a[i];
```

```
225         while (k > a[j - 1] && j)
226         {
227             a[j--] = a[j - 1];
228         }
229         a[j]=k;
230     }
231     b = (int*)malloc(n * sizeof(int*));
232     for (i = 0; i < n; i++)
233     {
234         scanf_s("%d", &b[i]);
235         j = i;
236         k = b[i];
237         while (k > b[j - 1] && j)
238         {
239             b[j--] = b[j - 1];
240         }
241         b[j] = k;
242     }
243     func(a, b, n);
244 }
245 void func(int* a, int* b, int n)
246 {
247     int i=0, j=0, k=0;
248     int* c;
249     c = (int*)malloc(n * 2 * 4);
250     while (i < n * 2)
251     {
252         while (a[j] >= b[k])
253         {
254             c[i++] = a[j++];
255         }
256         while (a[j] < b[k])
257         {
258             c[i++] = b[k++];
259         }
260     }
261     for (i = 0; i < n * 2; i++)
262         printf("%d ", c[i]);
263 }
264 //--------------------------------билет 3.2-----------------------------------------------
265 char* revers(char* a, int in, int ik);
266 int main()
267 {
268     char* a;
269     int i=0,k=1;
270     a = (char*)malloc(1);
271     while ((a[i++] = getchar()) != '\n')
272         a= (char*)realloc(a, i + 1);
273     a[i - 1] = '\0';
274     int in=0, ik=0;
275     i = 0;
276     while (a[i])
277     {
278         while (a[i] == ' ')
279         {
```

```
280             i++;
281             in = i;
282         }
283         while (a[i] != ' ' && a[i])
284         {
285             ik = i;
286             i++;
287         }
288         if (k % 2 != 0)
289         {
290             revers(a, in, ik);
291         }
292         k++;
293     }
294     puts(a);
295 }
296 char* revers(char* a, int in, int ik)
297 {
298     int i;
299     while (in < ik)
300     {
301         char c;
302         c = a[in];
303         a[in++] = a[ik];
304         a[ik--] = c;
305     }
306     return a;
307 }
308 //-------------------------------------билет 4.1-------------------------------------
309 void fibon(int n);
310 int main()
311 {
312     int n;
313     scanf_s("%d", &n);
314     fibon(n);
315 }
316 void fibon(int n)
317 {
318     static int a=1, b=1, c;
319     while (n)
320     {
321         printf("%d ", a);
322         c = a + b;
323         a = b;
324         b = c;
325         return fibon(--n);
326     }
327 }
328 //---------------------------Билет 4.2-----------------
329 char* revers(char * a,int in, int ik);
330 int main()
331 {
332     char* a;
333     int i = 0,n,in,ik;
334     a = (char*)malloc(1);
```

```
335     while ((a[i++] = getchar()) != '\n')
336         a = (char*)realloc(a, i + 1);
337     a[i - 1] = '\0';
338     n = i - 2;
339     a=revers(a,0, n);
340     i = 0;
341     while (a[i] != ' ')
342         i++;
343     in = i;
344     a=revers(a,0, i - 1);
345     i = n;
346     while (a[i] != ' ')
347         i--;
348     ik = i;
349     a = revers(a, i + 1, n);
350     a = revers(a, in, ik);
351     puts(a);
352 }
353 char* revers(char* a,int in, int ik)
354 {
355     while (in < ik)
356     {
357         char c;
358         c = a[in];
359         a[in++] = a[ik];
360         a[ik--] = c;
361     }
362     return a;
363 }
364 //--------------------------------билет 5.1-------------------------------
365 int main()
366 {
367     int** a,i,j,k,n,m;
368     scanf_s("%d", &n);
369     scanf_s("%d", &m);
370     a = (int**)malloc(n * sizeof(int*));
371     for (i = 0; i < n; i++)
372     {
373         a[i] = (int*)malloc(m * sizeof(int));
374     }
375     for (i = 0; i < n; i++)
376     {
377         for (j = 0; j < m; j++)
378         {
379             scanf_s("%d", &a[i][j]);
380         }
381     }
382     i = 0;
383     int mini=1000, maxj=0,j1=0;
384     while (i < n)
385     {
386         for (j = 0; j < m; j++)
387         {
388             if (a[i][j] < mini)
389             {
390                 mini = a[i][j];
```

```
391                 j1 = j;
392             }
393         }
394         for (k = 0; k < n; k++)
395         {
396             if (a[k][j1]>maxj)
397             {
398                 maxj = a[k][j1];
399             }
400         }
401         if (mini ==maxj)
402         {
403             printf("%d ", mini);
404             break;
405         }
406         i++;
407     }
408 }
409 //------------------------------ билет 5.2-------------------------
410 void func(char*a,char* b,int i,int j,int n);
411 int main()
412 {
413     char* a,*b;
414     int i = 0,j,n;
415     scanf_s("%d", &n);
416     rewind(stdin);
417     a = (char*)malloc(1);
418     while ((a[i++] = getchar()) != '\n')
419         a = (char*)realloc(a, i + 1);
420     a[i - 1] = '\0';
421     b = (char*)malloc(1);
422     j = 0;
423     while ((b[j++] = getchar()) != '\n')
424         b = (char*)realloc(b, j + 1);
425     b[j - 1] = '\0';
426     func(a, b, i - 1, j - 1,n);
427 }
428 void func(char* a, char* b, int i, int j,int n)
429 {
430     int i1, j1,j2;
431     if (i > j)
432     {
433         b = (char*)realloc(b, i + j);
434         for (i1 = 0; i1 < i; i1++)
435         {
436             j1 = j+1+i1;
437             while (j1>n)
438             {
439                 b[j1--] = b[j1 - 1];
440             }
441         }
442         i1 = 0;
443         j2 = n+1;
444         while (a[i1])
445             b[j2++] = a[i1++];
446         puts(b);
```

```
447     }
448     else
449     {
450         a = (char*)realloc(a, i + j);
451         for (i1 = 0; i1 < j; i1++)
452         {
453             j1 = i + 1 + i1;
454             while (j1 > n)
455             {
456                 a[j1--] = a[j1 - 1];
457             }
458         }
459         i1 = 0;
460         j2 = n+1;
461         while (b[i1])
462             a[j2++] = b[i1++];
463         puts(a);
464     }
465 }
466 //------------------------------билет 6.1---------------------------------
467 /*int** trans(int** a,int n);
468 int main()
469 {
470     int** a,i,j,n;
471     scanf_s("%d", &n);
472     a = (int**)malloc(n * sizeof(int**));
473     for (i = 0; i < n; i++)
474         a[i] = (int*)malloc(n * sizeof(int*));
475     for (i = 0; i < n; i++)
476     {
477         for (j = 0; j < n; j++)
478         {
479             a[i][j] = rand() % 9 + 1;
480             printf("%d ", a[i][j]);
481         }
482         printf("\n");
483     }
484     a=trans(a,n);
485     for (i = 0; i < n; i++)
486     {
487         for (j = 0; j < n; j++)
488         {
489             printf("%d ", a[i][j]);
490         }
491         printf("\n");
492     }
493
494 }
495 int** trans(int** a,int n)
496 {
497     static int i, j;
498     if (i < n)
499     {
500         if (j < n-1-i)
501         {
502             int c;
```

```
503             c = a[i][j];
504             a[i][j] = a[n - 1 - j][n - 1 - i];
505             a[n - 1 - j][n - 1 - i] = c;
506             j++;
507             return trans(a, n);
508         }
509         j = 0;
510         i++;
511         return trans(a, n);
512     }
513     return a;
514 }
515 //-----------------------------------------------билет 6.2----------------------
516 char* func(char* a, int in, int ik);
517 int main()
518 {
519     char* a;
520     int i=0,in=0,ik=0;
521     a = (char*)malloc(1);
522     while ((a[i++] = getchar()) != '\n')
523         a = (char*)realloc(a, i + 1);
524     a[i - 1] = '\0';
525     i -= 2;
526     while (a[i]!=' ')
527     {
528         i--;
529         ik = i;
530     }
531     i--;
532     while (a[i] != ' ')
533     {
534         i--;
535         in = i;
536     }
537     a = func(a, in, ik);
538     puts(a);
539 }
540 char* func(char* a, int in, int ik)
541 {
542     static int i = ik;
543     if (in != ik)
544     {
545         if (a[i])
546         {
547             a[i++] = a[i + 1];
548             return func(a, in, ik);
549         }
550         ik--;
551         i = ik;
552         return func(a, in, ik);
553     }
554     return a;
555 }
556 //-----------------------------------------------билет 7.1----------------------
557 int main()
```

```
{
    int* a,n,m,k,i=0,dev;
    scanf_s("%d", &k);
    scanf_s("%d", &n);
    scanf_s("%d", &m);
    a = (int*)malloc(k * 4);
    for (i = 0; i < k; i++)
    {
        a[i] = rand() % 20 + 10;
        printf("%d ", a[i]);
    }
    int j;
    for (dev = n+m/ 2; dev > 0; dev /= 2)
    {
        int fl = 0;
        do {
            fl = 0;
            for (i=n,j = i + dev; j <= m; i++, j++)
            {
                if (a[i] > a[j])
                {
                    int c;
                    c = a[i];
                    a[i] = a[j];
                    a[j] =  c;
                    fl = 1;
                }
            }

        } while (fl);
    }
    printf("\n");
    for (i = 0; i < k; i++)
    {
        printf("%d ", a[i]);
    }
}
//---------------------------------------билет 7.2---------------
char* func(char* a,int,int);
int main()
{
    char* a;
    int i=0;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a =(char*) realloc(a, i + 1);
    a[i - 1] = '\0';
    i = 0;
    int in=0, ik=0;
    while (a[i])
    {
        while (a[i] == ' ')
        {
            in = ++i;
        }
```

```
614         while (a[i] != ' ' && a[i])
615         {
616             ik = i++;
617         }
618         a = func(a, in, ik);
619     }
620     puts(a);
621 }
622 char* func(char* a,int in,int ik)
623 {
624     if (in<ik)
625     {
626         char c;
627         c = a[in];
628         a[in++] = a[ik];
629         a[ik--] = c;
630         return func(a, in, ik);
631     }
632     return a;
633 }
634 //---------------------------------билет 8.2---------------
635 void func(char* a);
636 int main()
637 {
638     char* a;
639     int i = 0;
640     a = (char*)malloc(1);
641     printf("Vvedite vashu stroku \n");
642     while ((a[i++] = getchar()) != '\n')
643         a = (char*)realloc(a, i + 1);
644     a[i - 1] = '\0';
645     func(a);
646 }
647 void func(char* a)
648 {
649     int i=0,length=0,j;
650     while (a[i])
651         length = i++;
652     for (i = 0; i < length-1; i++)
653     {
654         for (j = 0; j < length; j++)
655         {
656             if (a[j] > a[j + 1])
657             {
658                 char c;
659                 c = a[j];
660                 a[j] = a[j + 1];
661                 a[j + 1] = c;
662             }
663         }
664     }
665     puts(a);
666 }
667 //---------------------------------билет 9.1--------------------
668 int main()
669 {
```

```
670     int** a, n, i, j, m,k;
671     scanf_s("%d", &n);
672     scanf_s("%d", &m);
673     scanf_s("%d", &k);
674     a = (int**)malloc(n * sizeof(int**));
675     for (i = 0; i < n; i++)
676         a[i] = (int*)malloc(m * sizeof(int*));
677     for (i = 0; i < n; i++)
678     {
679         for (j = 0; j < m; j++)
680         {
681             a[i][j] = rand() % 20 + 10;
682             printf("%d ", a[i][j]);
683         }
684         printf("\n");
685     }
686     printf("\n");
687     for (i = 1; i < m; i++)
688     {
689         j = i;
690         while (a[k][i]< a[k][j - 1] && j-1>=0)
691         {
692             a[k][j] = a[k][j - 1];
693             j--;
694         }
695         a[k][j] = a[k][i];
696     }
697     for (i = 0; i < n; i++)
698     {
699         for (j = 0; j < m; j++)
700         {
701             printf("%d ", a[i][j]);
702         }
703         printf("\n");
704     }
705 }
706 //-------------------------------------билет 9.2---------------------------------
707 void func(char* a);
708 int main()
709 {
710     char* a;
711     int i = 0;
712     a = (char*)malloc(1);
713     printf("Vvedite vashu stroku \n");
714     while ((a[i++] = getchar()) != '\n')
715         a = (char*)realloc(a, i + 1);
716     a[i - 1] = '\0';
717     func(a);
718 }
719 void func(char* a)
720 {
721     int i=0, in=0, ik=0,c=0;
722     int in1, ik1;
723     while (c!=2)
724     {
725         while (a[i]==' ')
```

```
726         {
727             in = ++i;
728         }
729         while (a[i]!=' ' && a[i])
730         {
731             ik = i++;
732         }
733         in1 = in;
734         ik1 = ik;
735         while (in1 < ik1)
736         {
737             char c;
738             c = a[in1];
739             a[in1++] = a[ik1];
740             a[ik1--] = c;
741         }
742         c++;
743     }
744     in = 0;
745     while (in < ik)
746     {
747         char c;
748         c = a[in];
749         a[in++] = a[ik];
750         a[ik--] = c;
751     }
752     puts(a);
753 }
754 //----------------------------билет 10.1-------------------------------
755 void func(int,double);
756 int main()
757 {
758     int n;
759     double numb;
760     scanf_s("%lf", &numb);
761     scanf_s("%d", &n);
762 
763     func(n,numb);
764 }
765 void func(int n,double numb)
766 {
767     int over,k,s,i=0,j,i1,drob1;
768     float drob;
769     char* a;
770     a =(char*)malloc(1);
771     over = numb;
772     drob = numb - over;
773     while (over)
774     {
775         k = over / n;
776         s = over - n * k;
777         if (s < 10)
778         {
779             a[i] = '0' + s;
780         }
781         else
```

```
782         {
783             a[i] = 'A' + s-10;
784         }
785         i++;
786         a = (char*)realloc(a, i + 1);
787         over = k;
788     }
789     a[i] = '\0';
790     i1 = i;
791     i--;
792     j = 0;
793     while (j < i)
794     {
795         char c;
796         c = a[j];
797         a[j++] = a[i];
798         a[i--] = c;
799     }
800     a[i1++] = '.';
801     a = (char*)realloc(a, i1 + 10);
802     k = 0;
803     while (drob && k<10)
804     {
805         drob *= n;
806         drob1 = drob;
807         drob -= drob1;
808         if (drob1 < 10)
809         {
810             a[i1++] = '0' + drob1;
811         }
812         else
813         {
814             a[i1++] = 'A' + drob1 - 10;
815         }
816         k++;
817     }
818     a[i1] = '\0';
819     puts(a);
820 }
821 //-----------------------билет 10.2---------------------------------
822 void func(char** a, int n);
823 int main()
824 {
825     char** a;
826     int i, j,n;
827     scanf_s("%d", &n);
828     a = (char**)malloc(n * sizeof(char**));
829     for (i = 0; i < n; i++)
830         a[i] = (char*)malloc(1 * sizeof(char*));
831     rewind(stdin);
832     j = 0;
833     for (i = 0; i < n; i++)
834     {
835         rewind(stdin);
836         while ((a[i][j++] = getchar()) != '\n')
837         {
```

```
838             a[i] = (char*)realloc(a[i], (j + 1)*sizeof(char*));
839         }
840         a[i][j-1] = '\0';
841         j = 0;
842     }
843     func(a,n);
844 }
845 void func(char** a, int n)
846 {
847     int b[33] = {0}, i,j=0,k;
848     for (i = 0; i < n; i++)
849     {
850         while (a[i][j])
851         {
852             while (a[i][j] >= 'a' && a[i][j] <= 'z')
853             {
854                 b[a[i][j] - 'a']++;
855                 j++;
856             }
857             if(a[i][j])
858             j++;
859         }
860         printf("V %d stroke bukva :\n",i+1);
861         for (k = 0; k < 33; k++)
862         {
863             if (b[k] != 0)
864             {
865                 printf("%c vstrechaetsa %d raz \n", k + 'a', b[k]);
866             }
867         }
868         for (k = 0; k < 33; k++)
869             b[k] = 0;
870         j = 0;
871 
872     }
873 }
874 //-----------------------------------------------билет 11.1.....................
875 void func(int n,...);
876 int main()
877 {
878     func(2, 'i', 4, 1, 2, 3,4, 'f', 3, 1.1, 1.2, 0.3);
879 }
880 void func(int n,...)
881 {
882     va_list p;
883     int s=0,i,k;
884     double s1=0;
885     va_start(p, n);
886     while (n--)
887     {
888         switch (va_arg(p,char))
889         {
890             case 'i':
891             {
892                 k = va_arg(p, int);
893                 for (i = 0; i < k; i++)
```

```
                {
                    s = s + va_arg(p, int);
                }
            }break;
            case 'f':
            {
                k = va_arg(p,int);
                for (i = 0; i < k; i++)
                {
                    s1 = s1 + va_arg(p,double);
                }
            }break;
        }
    }
    printf("%d\n", s);
    printf("%lf \n", s1);
}
//--------------------------------билет 11.2-----------------
void func(int argc, char* argv);
int main(int argc,char*argv[])
{
    int i;
    for (i = 0; i < argc; i++)
        puts(argv[i]);
    func(argc,argv[1]);
}
void func(int argc, char* argv)
{
    int i = 0, j = 0,s=0,k=0,p=0;
    int in=0, ik=0,in1,ik1,length;
    char* b;
    b = (char*)malloc(1);
    while (argv[i])
    {
        while (argv[i] == ' ')
        {
            in = ++i;
        }
        while (argv[i] != ' ' && argv[i])
        {
            ik = ++i;
        }
        in1 = in;
        ik1 = ik;
        if (!p)
        {
            b = (char*)realloc(b, ik - in);
            length = ik - in;
            k = in;
            for (j = 0; j < length; j++)
                b[j] = argv[k++];
            b[j] = '\0';
        }
        j = 0;
        while (b[j] == argv[in] && b[j])
        {
```

```
950             s++;
951             j++; in++;
952         }
953         if (s == length)
954         {
955             while (in1<=ik1)
956             {
957                 j = ik1;
958                 while (argv[j])
959                 {
960                     argv[j++] = argv[j + 1];
961                 }
962                 ik1--;
963             }
964             i = 0;
965         }
966         s=in=ik= 0;
967         p++;
968     }
969     puts(argv);
970 }
971 //-----------------------------------билет 13.1-------------------------
972 int main()
973 {
974     int** a,i,j,k,n=7;
975     a = (int**)malloc(n * sizeof(int**));
976     for (i = 0; i < n; i++)
977         a[i] = (int*)malloc(n * sizeof(int*));
978     for (i = 0; i < n; i++)
979     {
980         for (j = 0; j < n; j++)
981         {
982             a[i][j] = rand() % 40 + 10;
983             printf("%d ", a[i][j]);
984         }
985         printf("\n");
986     }
987     for (i = 1; i < n; i++)
988     {
989         k = a[i][i];
990         j = i;
991         while (j && k > a[j - 1][j - 1])
992         {
993             a[j][j] = a[j - 1][j - 1];
994             j--;
995         }
996         a[j][j] = k;
997     }
998     printf("\n");
999     printf("\n");
1000     for (i = 0; i < n; i++)
1001     {
1002         for (j = 0; j < n; j++)
1003         {
1004             printf("%d ", a[i][j]);
1005         }
```

```
        printf("\n");
    }
}
//-----------------------------------билет 13.2-----------------------
void func(char* a);
int main()
{
    char* a;
    int i=0;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    func(a);
}
void func(char* a)
{
    int i=0,s=0,smax=0,k,n=0;
    while (a[i])
    {
        s = 0;
        k = 0;
        while (a[i] >= '1' && a[i] <= '9' && a[i])
        {
            s+= (a[i++] - '0');
            s *= 10;
        }
        s /=  10;
        if (s>smax)
        {
            smax = s;
        }
        if(a[i])
        i++;
    }
    printf("%d", smax);
}
//----------------------------билет 15-------------------------------------
char* func(double drob,int ss);
int main()
{
    int k,ss,i;
    char *a;
    double drob,n;
    scanf_s("%lf",&n);
    scanf_s("%d", &ss);
    k = n;
    drob = n - k;
    a=func(drob,ss);
    i = 0;
    while(a[i])
    {
        printf("%c", a[i++]);
    }
}
```

```
1062 char* func(double drob,int ss)
1063 {
1064     static int n;
1065     int celoe,i;
1066     double drob1;
1067     static char a[10];
1068     if (drob && n < 10)
1069     {
1070         drob1 = drob*ss;
1071         celoe = drob1;
1072         drob1 -= celoe;
1073         drob = drob1;
1074         (celoe>=10)?a[n] ='A'-10+celoe:a[n]=celoe+'0';
1075         n++;
1076         return func(drob, ss);
1077     }
1078     else
1079         return a;
1080 }
1081 //---------------------------------------билет 15.2.....................
1082 void func(char* a);
1083 int main()
1084 {
1085     char** a;
1086     int i, j,n;
1087     scanf_s("%d", &n);
1088     a = (char**)malloc(n * sizeof(char**));
1089     for (i = 0; i < n; i++)
1090     {
1091         a[i] = (char*)malloc(1*sizeof(char*));
1092     }
1093     j = 0;
1094     for (i = 0; i < n; i++)
1095     {
1096         rewind(stdin);
1097         while ((a[i][j++] = getchar()) != '\n')
1098         {
1099             a[i] = (char*)realloc(a[i], j + 1);
1100         }
1101         a[i][j - 1] = '\0';
1102         j = 0;
1103     }
1104     for (i = 0; i < n; i++)
1105     {
1106         func(a[i]);
1107     }
1108 }
1109 void func(char* a)
1110 {
1111     int i = 0,in=0,ik,s=0,j=0;
1112     while (a[i]!=' ' && a[i])
1113     {
1114         ik = i++;
1115     }
1116     i = 0;
1117     for (i = 0; i < 15; i++)
```

```
1118     {
1119         for (j = 0; j <= ik; j++)
1120         {
1121             if (a[j]=='a'+i)
1122             {
1123                 s++;
1124                 break;
1125             }
1126         }
1127     }
1128     if (s == 15)
1129     {
1130         for (i = 0; i <= ik; i++)
1131             printf("%c", a[i]);
1132         printf("\n");
1133     }
1134 }
1135 //-----------------------------------билет 16.1-----------------------------
1136 void func(int n, ...);
1137 int main()
1138 {
1139     char** a;
1140     int i, j, n;
1141     scanf_s("%d", &n);
1142     a = (char**)malloc(n * sizeof(char**));
1143     for (i = 0; i < n; i++)
1144     {
1145         a[i] = (char*)malloc(1 * sizeof(char*));
1146     }
1147     j = 0;
1148     for (i = 0; i < n; i++)
1149     {
1150         rewind(stdin);
1151         while ((a[i][j++] = getchar()) != '\n')
1152         {
1153             a[i] = (char*)realloc(a[i], j + 1);
1154         }
1155         a[i][j - 1] = '\0';
1156         j = 0;
1157     }
1158     func(n, a[0], a[1], a[2]);
1159 }
1160 void func(int n, ...)
1161 {
1162     int i=0,in=0, ik=0, length=0,
          maxlength=255,inmax,ikmax,k=0,k1,lengthstr=0,lengthstrmax;
1163     void*p=&n;
1164     void** p1,**p2=nullptr;
1165     while (n--)
1166     {
1167         p = (int*)p + 1;
1168         p1 = (void**)p;
1169         while (**(char**)p1)
1170         {
1171             while (**((char**)p1)==' ')
1172             {
```

```
                in = ++i;
                *p1 = (char*)*p1 + 1;
                lengthstr++;
            }
            while (**(char**)p1 != ' ' && (**(char**)p1))
            {
                ik = i++;
                *p1 = (char*)*p1 + 1;
                lengthstr++;
            }
            length = ik - in + 1;
            if (length < maxlength)
            {
                maxlength = length;
                inmax = in;
                ikmax = ik;
                k1 =k;
                p2 = (void**)p1;
            }
        }
        i =in=ik=0;
        if (k==k1)
            lengthstrmax = lengthstr;
        length = 0;
        k++;
        lengthstr = 0;
    }
    for (i=0; i<maxlength; i++)
    {
        printf("%c", *(*((char**)p2)-lengthstrmax+inmax+i));
    }
}
//--------------------билет 16.2-----------------------------------------------
char* swap(char* a, int in, int ik);
void func(char* a);
int main()
{
    char* a;
    int i = 0;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    func(a);
}
char* swap(char* a, int in, int ik)
{
    while (in < ik)
    {
        char c;
        c = a[in];
        a[in++] = a[ik];
        a[ik--] = c;
    }
    return a;
```

```
1229  }
1230  void func(char* a)
1231  {
1232      int i = 0, in = 0, ik = 0, minlength=255, length, inmax, ikmax;
1233      while (a[i])
1234      {
1235          while (a[i] == ' ')
1236          {
1237              in = ++i;
1238          }
1239          while (a[i] != ' ' && a[i])
1240          {
1241              ik = i++;
1242          }
1243          length = ik - in + 1;
1244          if (length < minlength)
1245          {
1246              minlength = length;
1247              inmax = in;
1248              ikmax = ik;
1249          }
1250      }
1251      int i1, i2=0;
1252      if (a[ikmax + 1])
1253      {
1254          i = ikmax+2;
1255          while (a[i] != ' ' && a[i])
1256              i1 = i++;
1257          a=swap(a, inmax, i1);
1258          i = inmax;
1259          while (a[i] != ' ')
1260              i1 = i++;
1261          a=swap(a, inmax, i1);
1262          i1 += 2;
1263          i++;
1264          while (a[i] != ' ' && a[i])
1265              i2 = i++;
1266          a=swap(a, i1, i2);
1267      }
1268      puts(a);
1269  }
1270  //----------------------------------------билет 17.1------------------------
1271  int func(int n)
1272  {
1273      static int i = 1,j=n;
1274      int a,b=i;
1275      if (n--)
1276      {
1277          printf("%d ", i);
1278          scanf_s("%d", &a);
1279          i++;
1280          if(func(n))
1281          printf("%d  -   %d\n",b,a);
1282      }
1283      else
1284      {
```

```
1285          return 1;
1286      }
1287  }
1288  int main()
1289  {
1290      int n;
1291      scanf_s("%d", &n);
1292      func(n);
1293  }
1294  //---------------------------------билет 17.2-------------------------
1295  void func(int n, ...)
1296  {
1297      int sumi=0;
1298      float sumf=0;
1299      va_list p ;
1300      va_start(p, n);
1301      while (n--)
1302      {
1303          char c;
1304          c = *p;
1305          switch (c)
1306          {
1307              case 'f':
1308              {
1309                  va_arg(p, char);
1310                  int i = *p, j;
1311                  va_arg(p, int);
1312                  for (j = 0; j < i; j++)
1313                      sumf += va_arg(p, double);
1314              }break;
1315              case 'i':
1316              {
1317                  va_arg(p, char);
1318                  int i = *p, j;
1319                  va_arg(p, int);
1320                  for (j = 0; j < i; j++)
1321                      sumi += va_arg(p, int);
1322              }break;
1323          }
1324      }
1325      printf("%d    %f", sumi, sumf);
1326  }
1327  int main()
1328  {
1329      int a, b, c;
1330      float x, y, z, q, w, e;
1331      scanf_s("%d%d%d", &a, &b, &c);
1332      scanf_s("%f%f%f%f%f%f", &x, &y, &z, &q, &w, &e);
1333      func(3, 'f', 3, x, y, z, 'i', 3, a, b, c, 'f', 3, q, w, e);
1334  }
1335  //---------------------------------------билет 18.1--------------------
1336  int main()
1337  {
1338      int** a;
1339      int i, j, k,n;
1340      scanf_s("%d", &n);
```

```
1341      a = (int**)malloc(n * sizeof(int*));
1342      for (i = 0; i < n; i++)
1343      {
1344          a[i] =(int*) malloc(n * sizeof(int));
1345      }
1346      for (i = 0; i < n; i++)
1347      {
1348          for (j = 0; j < n; j++)
1349          {
1350              a[i][j] = rand() % 20 + 10;
1351              printf("%d ", a[i][j]);
1352          }
1353          printf("\n");
1354      }
1355      int u,max=0;
1356      scanf_s("%d",&u);
1357      for (i = n - 1; i > n - u; i--)
1358      {
1359          max = a[u][i];
1360          k = i;
1361          for (j = i; j >= n - u; j--)
1362          {
1363              if (a[u][j] > max)
1364              {
1365                  max = a[u][j];
1366                  k = j;
1367              }
1368          }
1369          a[u][k] = a[u][i];
1370          a[u][i] = max;
1371      }
1372      for (i = 0; i < n; i++)
1373      {
1374          for (j = 0; j < n; j++)
1375          {
1376              printf("%d ", a[i][j]);
1377          }
1378          printf("\n");
1379      }
1380  }
1381  //-------------------------билет 18.2-----------------------------
1382  char* func(char *a,int  in,int ik);
1383  int  main()
1384  {
1385      char* a;
1386      int i = 0;
1387      a = (char*)malloc(1);
1388      printf("Vvedite vashu stroku \n");
1389      while ((a[i++] = getchar()) != '\n')
1390          a = (char*)realloc(a, i + 1);
1391      a[i - 1] = '\0';
1392      int in, ik=i-2;
1393      while (a[i] != ' ' && i>=0)
1394      {
1395          in = i--;
1396      }
```

```
1397      a=func(a, in, ik);
1398      puts(a);
1399  }
1400  char* func(char* a,int in,int ik)
1401  {
1402      if (in < ik)
1403      {
1404          char c;
1405          c = a[in];
1406          a[in++] = a[ik];
1407          a[ik--] = c;
1408          return func(a, in, ik);
1409      }
1410      return a;
1411  }
1412  //---------------------------------------билет 2
        19.1-----------------------------------------
1413  int main()
1414  {
1415      int*** a, i, j, k, n;
1416      scanf_s("%d", &n);
1417      a = (int***)malloc(n * sizeof(int**));
1418      if (a == nullptr)
1419          return 0;
1420      for (i = 0; i < n; i++)
1421      {
1422          a[i] = (int**)malloc(5 * sizeof(int*));
1423          if (a[i] == nullptr)
1424              return 0;
1425      }
1426      for (i = 0; i < n; i++)
1427      {
1428          for (j = 0; j < 5; j++)
1429          {
1430              a[i][j] = (int*)malloc(n * sizeof(int));
1431              if (a[i][j] == nullptr)
1432                  return 0;
1433          }
1434      }
1435      for (k = 0; k < n; k++)
1436      {
1437          for (i = 0; i < 5; i++)
1438          {
1439              for (j = 0; j < 5; j++)
1440              {
1441                  a[k][i][j] = rand() % 20 + 10;
1442                  printf("%d ", a[k][i][j]);
1443              }
1444              printf("\n");
1445          }
1446          printf("\n\n");
1447      }
1448      for (k = 0; k < n; k++)
1449      {
1450          for (i = 0; i < 5; i++)
1451          {
```

```
            for (j = 0; j <i; j++)
            {
                int f;
                f = a[k][i][j];
                a[k][i][j] = a[k][j][i];
                a[k][j][i] = f;
            }
        }
    }
    for (k = 0; k < n; k++)
    {
        for (i = 0; i < 5; i++)
        {
            for (j = 0; j < 5; j++)
            {
                printf("%d ", a[k][i][j]);
            }
            printf("\n");
        }
        printf("\n\n");
    }
}
//-----------------------билет 19.2-----------------
void func(char* a, char* b);
int main()
{
    char* a;
    int i = 0;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    rewind(stdin);
    char* b;
    i = 0;
    b = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((b[i++] = getchar()) != '\n')
        b = (char*)realloc(b, i + 1);
    b[i - 1] = '\0';
    func(a, b);
}
void func(char* a, char* b)
{
    int lengtha=0, lengthb=0,i=0;
    while (a[i])
    {
        lengtha=++i;
    }
    i = 0;
    while (b[i])
    {
        lengthb = ++i;
    }
    if (lengtha > lengthb)
```

```
1508     {
1509         b = (char*)realloc(b, lengtha + lengthb);
1510         int i1,i2,j=0;
1511         i2 = lengtha + lengthb;
1512         for (i1 = lengthb; i1 < i2; i1++,j++)
1513         {
1514             b[i1] = a[j];
1515         }
1516         b[i1] = '\0';
1517         printf("%d\n%d\n", lengtha, lengthb);
1518         puts(b);
1519     }
1520     else
1521     {
1522         a = (char*)realloc(a, lengtha + lengthb);
1523         int i1, i2, j = 0;
1524         i2 = lengtha + lengthb;
1525         for (i1 = lengtha; i1 < i2; i1++, j++)
1526         {
1527             a[i1] = b[j];
1528         }
1529         a[i1] = '\0';
1530         printf("%d\n%d\n", lengtha, lengthb);
1531         puts(a);
1532     }
1533 }
1534 //---------------------------------билет 20.1-------------------
1535 int main()
1536 {
1537     int** a, i, j, k,n;
1538     scanf_s("%d", &n);
1539     a = (int**)malloc(n * (sizeof(int)));
1540     for (i = 0; i < n; i++)
1541     {
1542         a[i] = (int*)malloc(n * sizeof(int));
1543     }
1544     for (i = 0; i < n; i++)
1545     {
1546         for (j = 0; j < n; j++)
1547         {
1548             scanf_s("%d", &a[i][j]);
1549         }
1550     }
1551     i = j = 0;
1552     k = 0;
1553     while (i < n)
1554     {
1555         int c = a[i][n-1-i];
1556         j = i + 1;
1557         while (j < n)
1558         {
1559             if (a[j][n - 1 - j] == c)
1560             {
1561                 a[j][n - 1 - j] = a[i][n - 1 - i] = 0;
1562             }
1563             j++;
```

```
        }
        j = 0;
        i++;
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}
//-------------------------билет 20.2-----------------------------
void func(char *a);
int main()
{
    char* a;
    int i = 0;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    func(a);
}
void func(char* a)
{
    int i = 0, j = 0, in = 0, ik = 0,n=0;
    while (a[i])
    {
        while (a[i] == ' ')
        {
            j = i;
            if (n)
            {
                while (a[j + 1] == ' ')
                {
                    int k;
                    k = j;
                    while (a[k])
                    {
                        a[k++] = a[k + 1];
                    }
                }
            }
            else
            {
                while (a[j] == ' ')
                {
                    int k;
                    k = j;
                    while (a[k])
                    {
                        a[k++] = a[k + 1];
                    }
```

```
                }
            }
            i++;
        }
        while (a[i] != ' ' && a[i])
        {
            i++;
            n++;
        }
    }
    while (a[i] != ' ')
        in=i--;
    while (a[in - 1] == ' ')
    {
        j = --in;
        while (a[j])
        {
            a[j] = a[j + 1];
        }
        in--;
    }
    puts(a);
}
//----------------------билет 21.1----------------------
void func(int** a,int n);
int main()
{
    int** a, i, j,n;
    scanf_s("%d", &n);
    a = (int**)malloc(n * sizeof(int*));
    for (i = 0; i < n; i++)
    {
        a[i] = (int*)malloc(n * sizeof(int));
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            a[i][j] = rand() % 20 + 10;
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    func(a,n);
}
void func(int** a,int n)
{
    int i=0, j, k,left=0,right=n-1,flag=1;
    while (i < n)
    {
        if (a[i][0] % 2==0)
        {
            while (left < right && flag)
            {
                flag = 0;
                for (j = left; j < right; j++)
```

```
1676                {
1677                    if (a[i][j] > a[i][j + 1])
1678                    {
1679                        int c;
1680                        c = a[i][j];
1681                        a[i][j] = a[i][j + 1];
1682                        a[i][j + 1] = c;
1683                        flag = 1;
1684                    }
1685                }
1686                right--;
1687                for (j = right; j > left; j--)
1688                {
1689                    if (a[i][j] < a[i][j - 1])
1690                    {
1691                        int c;
1692                        c = a[i][j];
1693                        a[i][j] = a[i][j - 1];
1694                        a[i][j - 1] = c;
1695                        flag = 1;
1696                    }
1697                }
1698                left++;
1699            }
1700            i++;
1701            left = 0;
1702            right = n - 1;
1703            flag = 1;
1704        }
1705        else
1706            i++;
1707    }
1708    for (i = 0; i < n; i++)
1709    {
1710        for (j = 0; j < n; j++)
1711        {
1712            printf("%d ", a[i][j]);
1713        }
1714        printf("\n");
1715    }
1716 }
1717 //-------------------------билет 21.2-----------------------
1718 char* swap(char* a, int in, int ik);
1719 void func(char* a);
1720 int main()
1721 {
1722     char* a;
1723     int i = 0;
1724     a = (char*)malloc(1);
1725     printf("Vvedite vashu stroku \n");
1726     while ((a[i++] = getchar()) != '\n')
1727         a = (char*)realloc(a, i + 1);
1728     a[i - 1] = '\0';
1729     func(a);
1730 }
1731 char* swap(char* a, int in, int ik)
```

```
{
    while (in < ik)
    {
        char c;
        c = a[in];
        a[in++] = a[ik];
        a[ik--] = c;
    }
    return a;
}
void func(char* a)
{
    int i = 0, in = 0, ik = 0, minlength = 255, length, inmax, ikmax;
    while (a[i])
    {
        while (a[i] == ' ')
        {
            in = ++i;
        }
        while (a[i] != ' ' && a[i])
        {
            ik = i++;
        }
        length = ik - in + 1;
        if (length < minlength)
        {
            minlength = length;
            inmax = in;
            ikmax = ik;
        }
    }
    int i1, i2 = 0;
    if (inmax - 1 > 0)
    {
        i1 = inmax - 2;
        while (a[i1] != ' ' && i1>=0)
            i1--;
        swap(a, i1 + 1, ikmax);
        i2 = i1 + 1;
        while (a[i2] != ' ')
            i2++;
        swap(a, i1+1, i2 - 1);
        i2++;
        swap(a, i2, ikmax);
    }
    puts(a);
}
//-------------------------билет 22.2-----------------------------------------
char* swap(char* a, int in, int ik);
void func(char* a,int n);
int main()
{
    char* a;
    int i = 0,n;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
```

```
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    scanf_s("%d", &n);
    func(a,n);
}
char* swap(char* a, int in, int ik)
{
    while (in < ik)
    {
        char c;
        c = a[in];
        a[in++] = a[ik];
        a[ik--] = c;
    }
    return a;
}
void func(char* a, int n)
{
    int i = 0, in = 0, ik = 0, length, inn, ikn;
    while (a[i] && n)
    {
        n--;
        while (a[i] == ' ')
        {
            in = ++i;
        }
        while (a[i] != ' ' && a[i])
        {
            ik = i++;
        }
        inn = in;
        ikn = ik;
    }
    int i1, i2 = 0;
    if (a[ikn + 1])
    {
        i1 = ikn + 2;
        while (a[i1] != ' ' && a[i1])
            i1++;
        swap(a, inn, i1 - 1);
        i1--;
        i2 = i1;
        while (a[i1] != ' ')
            i1--;
        swap(a, i1 + 1, i2);
        swap(a, inn, i1 - 1);
    }
    puts(a);
}
//-----------------------------------билет 24.1-----
int** func(int** a, int n);
int main()
{
    int** a, i, j, n;
    scanf_s("%d", &n);
```

```
1844      a = (int**)malloc(n * sizeof(int*));
1845      for (i = 0; i < n; i++)
1846      {
1847          a[i] = (int*)malloc(n * sizeof(int));
1848      }
1849      for (i = 0; i < n; i++)
1850      {
1851          for (j = 0; j < n; j++)
1852          {
1853              a[i][j] = rand() % 20 + 10;
1854              printf("%d ", a[i][j]);
1855          }
1856          printf("\n");
1857      }
1858      printf("\n");
1859      printf("\n");
1860      a=func(a, n);
1861      for (i = 0; i < n; i++)
1862      {
1863          for (j = 0; j < n; j++)
1864          {
1865              printf("%d ", a[i][j]);
1866          }
1867          printf("\n");
1868      }
1869  }
1870  int** func(int** a, int n)
1871  {
1872      static int i;
1873      if (i<n/2)
1874      {
1875          int c;
1876          c = a[i][n - 1 - i];
1877          a[i][n-1-i] = a[n-1-i][i];
1878          a[n-1-i][i] = c;
1879          i++;
1880          return func(a, n);
1881      }
1882      return a;
1883  }
1884  //--------------билет 24.2--------------------
1885  int func(char* a, char* b);
1886  int main(){
1887      char** a;
1888      int i, j, n;
1889      scanf_s("%d", &n);
1890      a = (char**)malloc(n * sizeof(char**));
1891      for (i = 0; i < n; i++)
1892      {
1893          a[i] = (char*)malloc(1 * sizeof(char*));
1894      }
1895      j = 0;
1896      for (i = 0; i < n; i++)
1897      {
1898          rewind(stdin);
1899          while ((a[i][j++] = getchar()) != '\n')
```

```
        {
        a[i] = (char*)realloc(a[i], j + 1);
        }
    a[i][j - 1] = '\0';
    j = 0;
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n-1; j++)
        {
            if (func(a[j], a[j + 1]))
            {
                char* p;
                p = a[j];
                a[j] = a[j + 1];
                a[j + 1] = p;
            }
        }
    }
    for (i = 0; i < n; i++)
        puts(a[i]);
}
int func(char* a, char* b)
{
    int i=0, j=0;
    while (a[i] == b[j])
    {
        i++;
        j++;
    }
    if (a[i] > b[j])
        return 1;
    else
        return 0;
}
//-----------------------------------билет 25.1-----------------------------
void func(int** a, int** b, int n, int n1, int m, int m1, int** c);
int main()
{
    int** a, ** b, ** c;
    int n, m, n1, m1, i, j;
    scanf_s("%d", &n);
    scanf_s("%d", &m);
    scanf_s("%d", &n1);
    scanf_s("%d", &m1);
    a = (int**)malloc(n * sizeof(int**));
    b = (int**)malloc(n1 * sizeof(int**));
    c = (int**)malloc(n * sizeof(int**));
    for (i = 0; i < n; i++)
    {
        a[i] = (int*)malloc(m * sizeof(int*));
        c[i] = (int*)malloc(m1 * sizeof(int*));
    }
    for (i = 0; i < n1; i++)
        b[i] = (int*)malloc(m1 * sizeof(int*));
    for (i = 0; i < n; i++)
```

```
    {
        for (j = 0; j < m; j++)
        {
            a[i][j] = rand() % 10 + 1;
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("\n");
    for (i = 0; i < n1; i++)
    {
        for (j = 0; j < m1; j++)
        {
            b[i][j] = rand() % 10 + 1;
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m1; j++)
        {
            c[i][j] = 0;
        }
    }
    func(a, b, n, n1, m, m1, c);
}
void func(int** a, int** b, int n, int n1, int m, int m1, int** c)
{
    int i = 0, j = 0, k, s, i1 = 0, j1 = 0, i2 = 0, j2 = 0;
    do
    {
        do
        {
            do
            {
                c[i][j] += a[i1][j1] * b[i2][j2];
                j1++;
                i2++;
            } while (j1 < m);
            i2 = 0;
            j1 = 0;
            j2++;
            j++;
        } while (j < m1);
        i2 = j2 = j1 = 0;
        i1++;
        i++;
        j = 0;
    } while (i < n);
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m1; j++)
        {
            printf("%d ", c[i][j]);
```

```
        }
        printf("\n");
    }
}
//---------------------------------билет 25.2--------------------------------
int** func(char* a);
int main()
{
    char* a;
    int i = 0;
    int* b,**n,k;
    a = (char*)malloc(1);
    printf("Vvedite vashu stroku \n");
    while ((a[i++] = getchar()) != '\n')
        a = (char*)realloc(a, i + 1);
    a[i - 1] = '\0';
    n=func(a);
    for (i = 0; i<n[1][0]; i++)
    {
        printf("%d ", n[0][i]);
    }
}
int** func(char* a)
{
    int i = 0, s = 0,k=1,j=0;
    int* b,**c;
    b = (int*)malloc(1 * sizeof(int));
    c = (int**)malloc(2 * sizeof(int*));
    while (a[i])
    {
        s = 0;
        while (a[i] >= '0' && a[i] <= '9' && a[i])
        {
            s = s * 10 + (a[i] - '0');
            i++;
        }
        if (s)
        {
            k++;
            b = (int*)realloc(b, k * sizeof(int));
            b[j] = s;
            j++;
        }
        if (a[i])
            i++;
    }
    c[0] =(int*)malloc(k*sizeof(int));
    for (i = 0; i < k-1; i++)
        c[0][i] = b[i];
    k--;
    c[1]=(int*)malloc(1*sizeof(int));
    c[1][0] = k;
    return c;
}
//----------------------------билет 26.1----------------------------
void func(int *a,int* b,int n,int n1);
```

```
int main()
{
    int* a, * b;
    int i,n,n1;
    scanf_s("%d", &n);
    scanf_s("%d", &n1);
    a = (int*)malloc(n * sizeof(int));
    b = (int*)malloc(n1 * sizeof(int));
    for (i = 0; i < n; i++)
    {
        int c;
        do
        {
            rewind(stdin);
            c = scanf_s("%d", &a[i]);
        } while (c && a[i]<a[i-1]);
    }
    for (i = 0; i < n1; i++)
    {
        int c;
        do
        {
            rewind(stdin);
            c = scanf_s("%d", &b[i]);
        } while (c && b[i] > b[i - 1] && i>0);
    }
    //a vozrast,b ubiv
    func(a, b, n, n1);
}
void func(int* a, int* b, int n, int n1)
{
    int i=0, j,j1;
    int* c;
    c = (int*)malloc((n + n1) * sizeof(int));
    j = 0;
    j1 = n1-1;
    while (i<n+n1)
    {
        while (a[j] <= b[j1] && j<n)
        {
            c[i++] = a[j++];
        }
        while (a[j] >= b[j1] && j1>=0)
        {
            c[i++] = b[j1--];
        }
        while (j == n && j1>=0)
        {
            c[i++] = b[j1--];
        }
        while (j1 < 0 && j<n)
        {
            c[i++] = a[j++];
        }
    }
    for (i = 0; i < n + n1; i++)
```

```
2124          printf("%d ", c[i]);
2125  }
2126  //-------------------------диалоговое окно билет 39.1---------------
2127  void fun1();
2128  void fun2();
2129  void fun3();
2130  void fun4();
2131  int main()
2132  {
2133      setlocale(LC_ALL, "RU");
2134      void(*p1[4])();
2135      int n;
2136      p1[0] = fun1;
2137      p1[1] = fun2;
2138      p1[2] = fun3;
2139      p1[3] = fun4;
2140      do {
2141          printf("Fun 1 - 1\n");
2142          printf("Fun 2 - 2\n");
2143          printf("Fun 3 - 3\n");
2144          printf("Fun 4 - 4\n");
2145          printf("Exit - 0\n");
2146          int c;
2147          do {
2148              rewind(stdin);
2149              printf("Выберите функцию\n");
2150              c = scanf_s("%d", &n);
2151          } while (c && (n < 0 || n>4));
2152          if (n)
2153              p1[n - 1]();
2154          else
2155              printf("Досвидания");
2156      } while (n);
2157  }
2158  void fun1()
2159  {
2160      printf("Лиза добрячок\n");
2161  }
2162  void fun2()
2163  {
2164      printf("Дима злюся\n");
2165  }
2166  void fun3()
2167  {
2168      printf("Eto func3 a Kirill LOH\n");
2169  }
2170  void fun4()
2171  {
2172      printf("Eto func4 a Mitja Loh\n");
2173  }*/
```

```
                        40(1)----------------------
//void fun(int* ms, int l, int r)
//{
//  int i = l, j = r, sr = ms[(l + r) / 2], t;
//  do {
//      while (ms[i] > sr) i++;
//      while (ms[j] < sr) j--;
//      if (i <= j) {
//          t = ms[i];
//          ms[i] = ms[j];
//          ms[j] = t;
//          i++;
//          j--;
//      }
//  } while (i <= j);
//  if (i < r)
//      fun(ms, i, r);
//  if (j > l)
//      fun(ms, l, j);
//}
//
//
//int main()
//{
// srand(time(NULL));
//  int* ms, i, s;
//  scanf_s("%i", &s);
// ms = (int*)malloc(s * sizeof(int));
//  for (i = 0; i < s; i++)
//      ms[i] = rand() % 15 + 1;
//  for (i = 0; i < s; i++)
//      printf("%i ", ms[i]);
//  fun(ms, 0, s - 1);
//  printf("\n");
//  for (i = 0; i < s; i++)
//      printf("%i ", ms[i]);
// return 0;
//}

////----------------------------40(2)-----------------------------
//char* revers(char* a, int in, int ik);
//int main()
//{
//  char* a;
//  int i = 0, k = 1;
//  a = (char*)malloc(1);
//  while ((a[i++] = getchar()) != '\n')
//      a = (char*)realloc(a, i + 1);
```

```
57  //  a[i - 1] = '\0';
58  //  int in = 0, ik = 0;
59  //  i = 0;
60  //  while (a[i])
61  //  {
62  //      while (a[i] == ' ')
63  //      {
64  //          i++;
65  //          in = i;
66  //      }
67  //      while (a[i] != ' ' && a[i])
68  //      {
69  //          ik = i;
70  //          i++;
71  //      }
72  //      revers(a, in, ik);
73  //      k++;
74  //  }
75  //  puts(a);
76  //}
77  //char* revers(char* a, int in, int ik)
78  //{
79  //  int i;
80  //  while (in < ik)
81  //  {
82  //      char c;
83  //      c = a[in];
84  //      a[in++] = a[ik];
85  //      a[ik--] = c;
86  //  }
87  //  return a;
88  //}
89
90  /// -------------------------39(1)-----------------------
91  //// использование массива указателей на функции на примере
92  //// организации диалогового меню
93
94  //void funk1();
95  //void funk2();
96  //void funk3();
97  //void(*spisok[])() =        // объявляется и инициализируется
98  //    {funk1, funk2, funk3}; // массив указателей на функции
99  //int menu();
100 //
101 //int main()
102 //{
103 //  setlocale(LC_ALL, "Russian");
104 //  int i;
105 //  puts("начало работы");
106 //  Sleep(5000);
107 //  do
108 //  {
109 //      i = menu();              // выбор номера из списка
110 //      if (i<0 || i>2) break;
111 //      spisok[i]();          // вызов i-ой функции через указатель на нее
112 //       /*(*spisok[i])();*/      // верно
```

```
113 //      // *spisok[i]();         // ошибка
114 //  } while (i >= 0 && i<3);
115 //  puts("окончание работы");
116 //  Sleep(5000);
117 //}
118 //int menu()
119 //{
120 //  char c;
121 //  int i;
122 //  do
123 //  {
124 //      system("CLS");
125 //      puts("1    funk1");
126 //      puts("2    funk2");
127 //      puts("3    funk3");
128 //      puts("0    выход");
129 //      c = (char)getchar();   // можно без преобразования к char
130 //  } while (!strchr("0123", c));
131 //  return c - 49;           // возвращаем выбранный номер
132 //}
133 //
134 //void funk1()
135 //{
136 // puts("выполняется функция   funk1");
137 //  Sleep(5000);
138 //}
139 //void funk2()
140 //{
141 // puts("выполняется функция   funk2");
142 //  Sleep(5000);
143 //}
144 //void funk3()
145 //{
146 // puts("выполняется функция   funk3");
147 //  Sleep(5000);
148 //}
149
150
151 ///// ________________________39(2)_____________________________________
152 //void fun(char* s1, char* s2)
153 //{
154 //  while (*s2 != '\n') {
155 //      *s1 = *s2;
156 //      s1++;
157 //      s2++;
158 //  }
159 //}
160 //
161 //int main()
162 //{
163     //char* str;
164     //int i= 0, count = 2, i1, j1;
165     //str = (char*)malloc(1);
166     //while ((str[i] = (char)getchar()) != '\n')
167     //  str = (char*)realloc(str, i++ + 2);
168     //str[i] = '\0';
```

```
169 //  i = 0;
170 //  while (str[i] && count--) {
171 //      while (str[i] == ' ') i++;
172 //      i1 = i;
173 //      if (!str[i])
174 //          return 0;
175 //      while (str[i] && str[i] != ' ') i++;
176 //      j1 = i;
177 //  }
178 //  fun(&str[i1], &str[j1]);
179 //  puts(str);
180 //  return 0;
181 //}
182
183 //// ---------------------37(1)------------------
184
185 //#include <stdio.h>
186 //#include <math.h>
187 //#include <conio.h>
188 //double f(double);
189 //double fun(double, double, double, double(*)(double));
190 //
191 //int main()
192 //{
193 //  setlocale(LC_ALL, "Russian");
194 //  double a, b, eps;
195 //  double(*ff)(double x);
196 //
197 //  puts("Введите границы a и b и точность вычисления корня");
198 //  scanf_s("%lf%lf%lf", &a, &b, &eps);
199 //  ff = f;
200 //  printf("a= %5.2lf  b=%5.2lf  корень = %6.4lf\n", a, b, fun(a, b, eps,      ↵
      ff));
201 //}
202 //double fun(double a, double b, double eps, double(*f1)(double))
203 //{
204 //  double fa, fb, c, fc;
205 //  do
206 //  {
207 //      fa = (*f1)(a);           // нахождение значений функции в концах
208 //      fb = (*f1)(b);           // выбранного интервала   [a,b]
209 //      c = (a + b) / 2.0;         // деление отрезка [a,b] пополам
210 //      fc = (*f1)(c);         // значение функции в середине отрезка
211 //      if (fa*fc>0) a = c;       //  выбор границ [c,b]
212 //      else if (fb*fc>0) b = c;  //  выбор границ [a,c]
213 //      else
214 //      {
215 //          puts("точный корень"); return c;
216 //      }
217 //  } while (fabs(a - b)>eps);  // пока не достигнута точность  eps
218 //  return c;
219 //}
220 //
221 //double f(double x)
222 //{
223 //  return (x*x - 3.);       // вычисление значения ф-ции f(x)=x^2-3 в точке x
```

```
224 //}                       // [ нахождение квадратного кореня из 3 ]
225
226
227 ////--------------------------37(2)-------------------------
228 //int fun(char* str)
229 //{
230 //  static int a, b;
231 //  if (*str) {
232 //      if (*str == '[')
233 //          a++;
234 //      else {
235 //          if (*str == ']') {
236 //              a--;
237 //              if (a < 0) {
238 //                  return -1;
239 //              }
240 //          }
241 //      }
242 //      fun(++str);
243 //  }
244 //  return a;
245 //}
246 //
247 //int main()
248 //{
249     //char* str;
250     //int i = 0;
251     //str = (char*)malloc(1);
252     //while ((str[i] = (char)getchar()) != '\n')
253     //  str = (char*)realloc(str, i++ + 2);
254     //str[i] = '\0';
255 //  if(!fun(str))
256 //      printf("Verno");
257 //  else
258 //      printf("NE Verno");
259 //  return 0;
260 //}
261
262 ////--------------------------36(1)-----------------------------
263 //int main()
264 //{
265 //  srand(time(NULL));
266 //  int** mt, i, j, n, t;
267 //  scanf_s("%i", &n);
268 //  mt = (int**)malloc(n * sizeof(int));
269 //  for (i = 0; i < n; i++) {
270 //      mt[i] = (int*)malloc(n * sizeof(int));
271 //  }
272 //  for (i = 0; i < n; i++) {
273 //      for (j = 0; j < n; j++) {
274 //          mt[i][j] = rand() % 15 + 1;
275 //      }
276 //  }
277 //  printf("#1.\n");
278 //  for (i = 0; i < n; i++) {
279 //      for (j = 0; j < n; j++) {
```

```
280 //          printf("%3i", mt[i][j]);
281 //      }
282 //      printf("\n");
283 //  }
284 //  i = 0;
285 //  do {
286 //      j = 0;
287 //      do {
288 //          t = mt[i][j];
289 //          mt[i][j] = mt[n - j - 1][n - i - 1];
290 //          mt[n - j - 1][n - i - 1] = t;
291 //          j++;
292 //
293 //      } while (j < n - i - 1);
294 //      i++;
295 //  } while (i < n - 1);
296 //  printf("\n#2.\n");
297 //  for (i = 0; i < n; i++) {
298 //      for (j = 0; j < n; j++) {
299 //          printf("%3i", mt[i][j]);
300 //      }
301 //      printf("\n");
302 //  }
303 //}
304
305 ////----------------------36(2)----------------------------
306 //void fun(char*);
307 //#define n 2
308 //int main()
309 //{
310     /*char** str;
311     int i = 0, j;
312     str = (char**)malloc(n * sizeof(char*));
313     for (i = 0; i < n; i++) {
314         str[i] = (char*)malloc(1);
315     }
316     for (i = 0; i < n; i++) {
317         j = 0;
318         while ((str[i][j] = (char)getchar()) != '\n')
319             str[i] = (char*)realloc(str[i], j++ + 2);
320         str[i][j] = '\0';
321     }*/
322     //  void(*foo)(char* x);
323     //  foo = fun;
324     //  for(i = 0; i < n; i++)
325     //      foo(str[i]);
326     //  for(i = 0; i < n; i++)
327     //      puts(str[i]);
328     //  return 0;
329     //}
330     //
331     //void fun(char* str)
332     //{
333     //  int i = 0, i1, j1 = 0;
334     //  while (str[i] == ' ') i++;
335     //  i1 = i;
```

```
336     //  while (str[i1]) {
337     //      str[j1] = str[i1];
338     //      i1++;
339     //      j1++;
340     //  }
341     //  str[j1] = '\0';
342     //  i = j1 - 1;
343     //  while (str[i] == ' ' && i >= 0) i--;
344     //  i1 = i + 1;
345     //  while (i1 < j1) {
346     //      str[j1 - 1] = str[j1];
347     //      j1--;
348     //  }
349     //  i = 0;
350     //  while (str[i]) {
351     //      while (str[i] && str[i] != ' ') i++;
352     //      i1 = i;
353     //      while (str[i] == ' ') i++;
354     //      j1 = i;
355     //      if (j1 - i1 > 1) {
356     //          i1++;
357     //          while (str[j1]) {
358     //              str[i1] = str[j1];
359     //              i1++;
360     //              j1++;
361     //          }
362     //          str[i1] = '\0';
363     //      }
364     //  }
365     //}
366
367
368     ////-------------------33(1)-------------------------
369     //int main()
370     //{
371     // srand(time(NULL));
372     //  int* ms, i, count = 0, n;
373     //  double srAr = 0, sum = 0;
374     //  scanf_s("%i", &n);
375     //  ms = (int*)malloc(n * sizeof(int));
376     //  for (i = 0; i < n; i++)
377     //      ms[i] = rand() % 15 + 1;
378     //  for (i = 0; i < n; i++)
379     //      printf("%i ", ms[i]);
380     //  for (i = 0; i < n; i++)
381     //      srAr += ms[i];
382     //  srAr /= n;
383     //  for (i = 0; i < n; i++)
384     //      if (ms[i] < srAr) {
385     //          count++;
386     //          sum += ms[i];
387     //      }
388     //  printf("\nsrAr = %lf\ncount = %i\nsum = %lf", srAr, count, sum);
389     //  return 0;
390     //}
391
```

```
392
393     ////-------------------------33(2)-------------------------
394     //int fun(char** s, int n)
395     //{
396     //  static char** p;
397     //  static int max, med, flag, num, maxnum, word;
398     //  if (n) {
399     //      if (!word) {
400     //          flag = 1;
401     //          p = s;
402     //      }
403     //      if (**p) {
404     //          if (**p == ' ') {
405     //              (*p)++;
406     //              flag = 1;
407     //              word = 0;
408     //              fun(s, n);
409     //          }
410     //          if (**p && **p != ' ') {
411     //              if(**p >= 'A' && **p <= 'Z' && flag == 1){
412     //                  med++;
413     //                  (*p)++;
414     //                  flag = 0;
415     //                  word = 1;
416     //                  fun(s, n);
417     //              }
418     //              else {
419     //                  (*p)++;
420     //                  flag = 0;
421     //                  word = 1;
422     //                  fun(s, n);
423     //              }
424     //          }
425     //      }
426     //      if (med >= max) {
427     //          max = med;
428     //          maxnum = num;
429     //      }
430     //      num++;
431     //      fun(++s, --n);
432     //  }
433     //  return maxnum;
434     //}
435     //
436     //
437     //int main(/*int argc, char* argv[]*/)
438     //{
439         //int argc = 3;
440         //char** argv;
441         //int i = 0, j;
442         //argv = (char**)malloc(argc * sizeof(char*));
443         //for (i = 0; i < argc; i++) {
444         //  argv[i] = (char*)malloc(1);
445         //}
446         //for (i = 0; i < argc; i++) {
447         //  j = 0;
```

```
448         //  while ((argv[i][j] = (char)getchar()) != '\n')
449         //      argv[i] = (char*)realloc(argv[i], j++ + 2);
450         //  argv[i][j] = '\0';
451         //}
452     //  for (i = 0; i < argc; i++) {
453     //      puts(argv[i]);
454     //  }
455     //  int num = fun(argv, argc);
456     //  printf("%i", num);
457     //  return 0;
458     //}
459
460     ////.....32(1)----------
461
462     //int fun(int** mt, int s)
463     //{
464     //  static int i = 1, j, count = 5, sum;
465     //  if (i < s) {
466     //      if (j < i) {
467     //          if (!(mt[i][j] % 2) && count) {
468     //              sum += mt[i][j];
469     //              count--;
470     //          }
471     //          j++;
472     //          fun(mt, s);
473     //      }
474     //      else {
475     //          i++;
476     //          j = 0;
477     //          fun(mt, s);
478     //      }
479     //  }
480     //  return sum;
481     //}
482     //
483     //int main()
484     //{
485         //srand(time(NULL));
486         //int** mt, s, i, j;
487         //scanf_s("%i", &s);
488         //mt = (int**)malloc(s * sizeof(int));
489         //for (i = 0; i < s; i++)
490         //  mt[i] = (int*)malloc(s * sizeof(int));
491         //for (i = 0; i < s; i++) {
492         //  for (j = 0; j< s; j++) {
493         //      mt[i][j] = rand() % 15 + 1;
494         //  }
495         //}
496         //for (i = 0; i < s; i++) {
497         //  for (j = 0; j < s; j++) {
498         //      printf("%3i", mt[i][j]);
499         //  }
500         //  printf("\n");
501         //}
502     //  printf("sum = %i", fun(mt, s));
503     //  return 0;
```

```
504     //}
505
506     ////----------------------32(2)---------------------------
507
508     //int fun(char* str)
509     //{
510     //  int i = 0, sum = 0;
511     //  while (str[i]) {
512     //      while (str[i] == ' ') i++;
513     //      while (str[i] && str[i] != ' ') {
514     //          while (str[i] >= '0' && str[i] <= '9') {
515     //              sum = sum * 10 + str[i] - '0';
516     //              i++;
517     //          }
518     //          break;
519     //      }
520     //  }
521     //  return (!(sum % 4)) ? sum : 0;
522     //}
523     //
524     //int main(int argc, char* argv[])
525     //{
526     //  int i = 0, sum = 0;
527     //  for (i = 1; i < argc; i++) {
528     //      sum += fun(argv[i]);
529     //  }
530     //  printf("sum = %i", sum);
531     //  return 0;
532     //}
533
534     //// ----------------------31(1)-----------------------
535     //void fun(int* ms, int s)
536     //{
537     //  int gap = s, i, j, flag, t;
538     //  while (gap) {
539     //      gap /= 2;
540     //      do {
541     //          flag = 0;
542     //          for (i = 0, j = i + gap; j < s; i++, j++) {
543     //              if (ms[i] < ms[j]) {
544     //                  t = ms[i];
545     //                  ms[i] = ms[j];
546     //                  ms[j] = t;
547     //                  flag = 1;
548     //              }
549     //          }
550     //
551     //      } while (flag);
552     //  }
553     //
554     //}
555     //
556     //int main()
557     //{
558     //  srand(time(NULL));
559     //  int* ms, i, s;
```

```
560     //  scanf_s("%i", &s);
561     //  ms = (int*)malloc(s * sizeof(int));
562     //  for (i = 0; i < s; i++)
563     //      ms[i] = rand() % 15 + 1;
564     //  printf("#1:");
565     //  for (i = 0; i < s; i++)
566     //      printf(" %i", ms[i]);
567     //  fun(ms, s);
568     //  printf("\n#2:");
569     //  for (i = 0; i < s; i++)
570     //      printf(" %i", ms[i]);
571     //  return 0;
572     //}
573
574     ////-----------------------31(2)----------------------
575
576     //void fun(char* str)
577     //{
578     //  static int i = 0, i1 = 0, j1 = 0, flag;
579     //  if (str[i]) {
580     //      if (j1 - i1 > 1 && flag) {
581     //          if (str[j1]) {
582     //              str[i1 + 1] = str[j1];
583     //              i1++;
584     //              j1++;
585     //          }
586     //          else {
587     //              str[i1 + 1] = '\0';
588     //              flag = 0;
589     //          }
590     //          fun(str);
591     //
592     //      }
593     //      if (str[i] && str[i] != ' ') {
594     //          i1 = i;
595     //          i++;
596     //          flag = 0;
597     //          fun(str);
598     //      }
599     //      if (str[i] == ' ') {
600     //          j1 = i;
601     //          i++;
602     //          flag = 1;
603     //          fun(str);
604     //      }
605     //  }
606     //}
607     //
608     //int main()
609     //{
610         //char* str;
611         //int i = 0;
612         //str = (char*)malloc(1);
613         //while ((str[i] = (char)getchar()) != '\n')
614         //  str = (char*)realloc(str, i++ + 2);
615         //str[i] = '\0';
```

```
616     //  i = 0;
617     //  while (str[i] == ' ') i++;
618     //  fun(&str[i]);
619     //  puts(str);
620     //  return 0;
621     //}
622
623     //// ----------------------30(1)-------------------------
624     //void fun(int* ms, int s)
625     //{
626     //  int i = 0, j;
627     //  while (i < s) {
628     //      printf("ms[%i] = ", i);
629     //      do {
630     //          j = scanf_s("%i", &ms[i]);
631     //          rewind(stdin);
632     //      } while (!j);
633     //      if (j && ((i == 0) || (ms[i] < ms[i - 1])))
634     //          i++;
635     //  }
636     //}
637     //
638     //int main()
639     //{
640     //  int* ms1, * ms2, i, j, k, * ms3, s1, s2;
641     //  scanf_s("%i %i", &s1, &s2);
642     //  ms1 = (int*)malloc(s1 * sizeof(int));
643     //  ms2 = (int*)malloc(s2 * sizeof(int));
644     //  ms3 = (int*)malloc((s1 + s2) * sizeof(int));
645     //  fun(ms1, s1);
646     //  fun(ms2, s2);
647     //  printf("#1:");
648     //  for (i = 0; i < s1; i++)
649     //      printf(" %i", ms1[i]);
650     //  printf("\n#2:");
651     //  for (i = 0; i < s2; i++)
652     //      printf(" %i", ms2[i]);
653     //  i = s1 - 1, j = s2 - 1, k = 0;
654     //  while (i >= 0 && j >= 0) {
655     //      if (ms1[i] <= ms2[j])
656     //          ms3[k++] = ms1[i--];
657     //      if (ms2[j] < ms1[i])
658     //          ms3[k++] = ms2[j--];
659     //  }
660     //  while (i >= 0)
661     //      ms3[k++] = ms1[i--];
662     //  while (j >= 0)
663     //      ms3[k++] = ms2[j--];
664     //  printf("\n#3:");
665     //  for (i = 0; i < s1 + s2; i++)
666     //      printf(" %i", ms3[i]);
667     //  free(ms1);
668     //  free(ms2);
669     //  free(ms3);
670     //  return 0;
671     //}
```

```
672
673
674     ////------------------------30(2)-------------------------
675
676     //void fun(char* s1, char* s2, int n, int len1, int len2)
677     //{
678     //  int i, j;
679     //  for (i = len2, j = len1 + len2; i >= n; i--, j--) {
680     //      s2[j] = s2[i];
681     //  }
682     //  for (i = n, j = 0; s1[j]; i++, j++) {
683     //      s2[i] = s1[j];
684     //  }
685     //}
686     //
687     //int main()
688     //{
689     //  char* str1,* str2, len1, len2, n;
690     //  int i = 0;
691     //  str1 = (char*)malloc(1);
692     //  while ((str1[i] = (char)getchar()) != '\n')
693     //  str1 = (char*)realloc(str1, i++ + 2);
694     //  str1[i] = '\0';
695     //  len1 = i;
696     //  i = 0;
697     //  str2 = (char*)malloc(1);
698     //  while ((str2[i] = (char)getchar()) != '\n')
699     //      str2 = (char*)realloc(str2, i++ + 2);
700     //  str2[i] = '\0';
701     //  len2 = i;
702     //  str2 = (char*)realloc(str2, len1 + len2);
703     //  printf("n = ");
704     //  scanf_s("%i", &n);
705     //  fun(str1, str2, n, len1, len2);
706     //  puts(str2);
707     //  return 0;
708     //}
709
710     ////------------------------29(1)-------------------------
711     //void fun(double num, int ss, int t)
712     //{
713     //  int cel;
714     //  printf("0.");
715     //  while (num && t) {
716     //      num *= ss;
717     //      cel = num;
718     //      num -= cel;
719     //      printf("%c", (cel > 9) ? cel - 10 + 'A' : cel + '0');
720     //      t--;
721     //  }
722     //}
723     //
724     //int main()
725     //{
726     //  double num;
727     //  int ss, t;
```

```
728     //  scanf_s("%lf", &num);
729     //  scanf_s("%i", &ss);
730     //  scanf_s("%i", &t);
731     //  fun((num - (int)num), ss, t);
732     //  return 0;
733     //}
734
735     ////------------------------29(2)-------------------------
736     //void fun(char* str)
737     //{
738     //  char t;
739     //  static int i, i1, j1, flag1, flag2;
740     //  if (str[i] == ' ' && !flag1) {
741     //      i++;
742     //      i1 = i;
743     //      fun(str);
744     //  }
745     //  if (str[i] && str[i] != ' ' && !flag2) {
746     //      i++;
747     //      j1 = i;
748     //      flag1 = 1;
749     //      fun(str);
750     //  }
751     //  if (str[j1]) {
752     //      flag2 = 1;
753     //      str[i1] = str[j1];
754     //      i1++;
755     //      j1++;
756     //      fun(str);
757     //  }
758     //  else {
759     //      str[i1] = '\0';
760     //  }
761     //}
762     //
763     //int main()
764     //{
765     //  char* str;
766     //  int i = 0;
767     //  str = (char*)malloc(1);
768     //  while ((str[i] = (char)getchar()) != '\n')
769     //      str = (char*)realloc(str, i++ + 2);
770     //  str[i] = '\0';
771     //  fun(str);
772     //  puts(str);
773     //  return 0;
774     //}
775
776
777     ////------------------------28(1)-------------------------
778     //void fun(int** mt, int s)
779     //{
780     //  static int i = 1, j;
781     //  int t;
782     //  if (i < s) {
783     //      if (j < i) {
```

```
784     //          t = mt[i][j];
785     //          mt[i][j] = mt[j][i];
786     //          mt[j][i] = t;
787     //          j++;
788     //          fun(mt, s);
789     //      }
790     //      else {
791     //          i++;
792     //          j = 0;
793     //          fun(mt, s);
794     //      }
795     //  }
796     //}
797     //
798     //int main()
799     //{
800     // srand(time(NULL));
801     //  int** mt, s, i, j;
802     //  scanf_s("%i", &s);
803     //  mt = (int**)malloc(s * sizeof(int));
804     //  for (i = 0; i < s; i++)
805     //      mt[i] = (int*)malloc(s * sizeof(int));
806     //  for (i = 0; i < s; i++) {
807     //      for (j = 0; j < s; j++) {
808     //          mt[i][j] = rand() % 15 + 1;
809     //      }
810     //  }
811     // printf("#1.\n");
812     //  for (i = 0; i < s; i++) {
813     //      for (j = 0; j < s; j++) {
814     //          printf("%3i", mt[i][j]);
815     //      }
816     //      printf("\n");
817     //  }
818     //  fun(mt, s);
819     // printf("\n#2.\n");
820     //  for (i = 0; i < s; i++) {
821     //      for (j = 0; j < s; j++) {
822     //          printf("%3i", mt[i][j]);
823     //      }
824     //      printf("\n");
825     //  }
826     //}
827
828     ////------------------------28(2)..............................
829
830     //char* fun(char** str, int n)
831     //{
832     //  int i, j, k, len = 0, max = 0, i1, j1, le, ri;
833     //  for (i = 0; i < n; i++) {
834     //      j = 0;
835     //      while (str[i][j]) {
836     //          while (str[i][j] == ' ') j++;
837     //          i1 = j;
838     //          while (str[i][j] && str[i][j] != ' ') j++;
839     //          j1 = j;
```

```
840     //          len = j1 - i1;
841     //          if (len > max || max == 0) {
842     //              max = len;
843     //              k = i;
844     //              le = i1;
845     //              ri = j1;
846     //          }
847     //      }
848     //  }
849     //  char* str1;
850     //  str1 = (char*)malloc(max);
851     //  for (i1 = 0, j1 = le; j1 < ri; i1++, j1++) {
852     //      str1[i1] = str[k][j1];
853     //  }
854     //  str1[i1] = '\0';
855     //  return str1;
856     //}
857     //
858     //int main()
859     //{
860     //  char** str;
861     //  int i = 0, j, n;
862     //  scanf_s("%i", &n);
863     //  rewind(stdin);
864     //  str = (char**)malloc(n * sizeof(char*));
865     //  for (i = 0; i < n; i++) {
866     //      str[i] = (char*)malloc(1);
867     //  }
868     //  for (i = 0; i < n; i++) {
869     //      j = 0;
870     //      while ((str[i][j] = (char)getchar()) != '\n')
871     //          str[i] = (char*)realloc(str[i], j++ + 2);
872     //      str[i][j] = '\0';
873     //  }
874     //  printf("\nMAX - %s", fun(str, n));
875     //  return 0;
876     //}
877
878     ////--------------------26(2)-------------------------
879
880     //void fun(char* str)
881     //{
882     //  int i = 0, j, flag = 1;
883     //  char a;
884     //  while (str[i]) {
885     //      while (str[i] == ' ') i++;
886     //      while (str[i] && str[i] != ' ') {
887     //          j = i + 1;
888     //          if (flag) {
889     //              a = str[i];
890     //              flag = 0;
891     //          }
892     //          if (str[i] == a) {
893     //              while (str[j]) {
894     //                  str[j - 1] = str[j];
895     //                  j++;
```

```
896     //              }
897     //              str[j - 1] = '\0';
898     //              i--;
899     //          }
900     //          i++;
901     //      }
902     //  }
903     //}
904     //
905     //int main()
906     //{
907     //  char* str;
908     //  int i = 0;
909     //  str = (char*)malloc(1);
910     //  while ((str[i] = (char)getchar()) != '\n')
911     //      str = (char*)realloc(str, i++ + 2);
912     //  str[i] = '\0';
913     //  fun(str);
914     //  puts(str);
915     //  return 0;
916     //}
```