

## **DXC - Essay Topics**

### **1. He who never learnt to obedient can't be a good leader. Justify this statement with both positive and negatives**

A leader is a person who is respected and respects others. A leader is a strong person who is a foundation for his or her followers. Leaders are not always to be the best or on top. Leaders need to be a base for the followers in order to accomplish what is best for the team. Leaders need to be role models for followers and need to encourage and build up the team they are working alongside.

A leader should have a power of intuition and taking precautionary steps in order to tame the challenging situations efficiently and successfully. Intuition, being one of the best characteristics of a leader, forecasts everything waited. It helps the respective leader to apprise public, in advance, a situation to cause its vigorous impact on the people perilously or constructively. In other words a good leader can arouse the sentiments against as well as in favour of individual, regime and power.

Managing, controlling and motivating/ persuading are the required characteristics of a leader. Good leader should control his emotions and anger in public. He should motivate the public through his words and deeds without causing an impression that it may hurt common person as well.

Some of the most significant characteristics of leaders are :

- ✓ **Exemplary character**. It is of utmost importance that a leader is trustworthy to lead others. A leader needs to be trusted and be known to live their life with honestly and integrity. A good leader “walks the talk” and in doing so earns the right to have responsibility for others. True authority is born from respect for the good character and trustworthiness of the person who leads.
- ✓ **Enthusiastic** : People will respond more openly to a person of passion and dedication. Leaders need to be able to be a source of inspiration, and be a motivator towards the required action or cause. Although the responsibilities and roles of a leader may be different, the leader needs to be seen to be part of the team working towards the goal. This kind of leader will not be afraid to roll up their sleeves and get dirty.
- ✓ **Confident**: In order to lead and set direction a leader needs to appear confident as a person and in the leadership role. Such a person inspires confidence in others and draws out the trust and best efforts of the team to complete the task well. A leader who

conveys confidence towards the proposed objective inspires the best effort from team members.

- ✓ **Tolerance:** Good leaders are tolerant of ambiguity and remain calm, composed and steadfast to the main purpose. Storms, emotions, and crises come and go and a good leader takes these as part of the journey and keeps a cool head.
- ✓ **Analytical Thinking :** Not only does a good leader view a situation as a whole, but is able to break it down into sub parts for closer inspection. While keeping the goal in view, a good leader can break it down into manageable steps and make progress towards it.
- ✓ **Excellence :** A good leader is committed to excellence. Second best does not lead to success. The good leader not only maintains high standards, but also is proactive in raising the bar in order to achieve excellence in all areas.

Overall, leadership involves being an effective communicator, being trustworthy, and being ambitious. The leader needs to be able to connect on some level with his followers. His followers need to be able to depend on him. Lastly, he needs the inner motivation to want to lead the group of people.

## 2. And the role of nuclear weaponry

**Definition :** **Arms race**, a pattern of competitive acquisition of military capability between two or more countries. The term is often used quite loosely to refer to any military buildup or spending increases by a group of countries. The competitive nature of this buildup often reflects an adversarial relationship. The arms race concept is also used in other fields. However, the discussion in this article is limited to military arms races.

Those technologies have now developed into nuclear weapons. The destructive power of a nuclear weapon is just one threat to consider. The radioactive fallout from such a weapon can cause long-term consequences for a population that was close to the explosion. Radiated soil, weapon debris, and radioactive particles falling from the sky are all threats that must be taken seriously, which means a nation with nuclear weapons provides a great threat to a potential invader.

If enough nuclear weapons are launched and allowed to detonate, however, then life as we know it on our planet will cease to exist. Any survivors would be forced to endure unthinkable conditions.

Here are the essential pros and cons of nuclear weapons to consider.

### **The Pros of Nuclear Weapons**

**1. It is a deterrent for starting a major conflict.**

Only a handful of nations are armed with nuclear weapons. The US and Russia have the most weapons, but France, China, the United Kingdom, and Pakistan all have more than 100 weapons. The presence of these weapons and their immense destructive capabilities is a major deterrent for starting a major conflict, like the two world wars that are fought in the first half of the 20th century. Even when both nations have nuclear weapons, their mutually assured destruction is enough to cause diplomacy to rule the day.

**2. It reinforces national borders.**

Our society is becoming ever-more global. Because of the internet, communications improvements, and other technological advances, we can talk with anyone, anywhere, with a simple data connection. New transportation technologies are in development that can provide long-distance transportation at speeds which were unthinkable just a generation ago. By reinforcing national borders, a global society of cooperation can be built through negotiation and diplomacy instead of warfare.

**3. It changes the power and status of a nation.**

Countries with nuclear weapons are treated different on the global stage than countries without those weapons. North Korea may only have a dozen or so nuclear weapons at best estimate, but their demands are taken with an extra level of seriousness and scrutiny because they have developed rudimentary nuclear weapon technology. Because there is a desire to avoid such a devastating conflict, most nations with nuclear weapons can generally get what they want on-demand.

**4. They can be fired without a direct threat to personnel.**

The nuclear weapons dropped in Japan were carried by bombers, staffed with a flight crew. Rockets can be fired remotely from stations and intercontinental ballistic technologies can let someone manage the weapon from thousands of miles away. Nuclear weapons do not require a battlefield presence in the same way a conventional weapon may require.

**5. Nuclear weapons can be positioned in a variety of locations.**

Submarines and naval craft can be outfitted with nuclear weapons so that they can be fired from any ocean-based location on the planet. Storage silos, buried underground, can fire ICBM-outfitted nuclear weapons. They can be fired from remote trucks, put into aircraft missiles, or still be dropped by bombers like they were in the 1940s. The versatility of positioning that a nuclear weapon offers is quite profound.

**6. They provide the foundations for other technologies.**

The technologies that have helped to create nuclear weapons have fostered other technologies that have benefited society in many ways. Nuclear power provides a relatively clean source of energy that is used to power hundreds of thousands of homes around the world. Nuclear reactors have been used to build naval vessels. Many types of nuclear-powered propulsion have been proposed for future spacecraft.

**7. It is a highly reliable technology.**

As a power source, nuclear energy can run uninterrupted for more than a year without disruption, even during difficult weather conditions. The refinement of a nuclear weapon is similar to this advantage. Once installed, the delivery platform reliability of the weapon is strong. Even if the warheads are not modernized, the surety of the weapon and its readiness make it a consistent threat that cannot be ignored.

## **The Cons of Nuclear Weapons**

### **1. They have added development and maintenance costs.**

According to the Congressional Budget Office, the United States spends an estimated \$34.8 billion per year to maintain, operate, and upgrade its nuclear weapons arsenal. Estimates from Nuclear Threat Initiative suggest the figures may be a little lower, but still costing the US up to \$25 billion per year. To put that in perspective, an investment into food products at that level would provide up to 75 billion meals for those living in poverty.

### **2. Detonation has a major ethical impact.**

Only two cities have ever been directly affected by nuclear weapons in history. Both were in Japan: Hiroshima and Nagasaki. Just two bombs caused the death of over 129,000 people, with many of them being civilians. Acute effects from the bombs may have contributed to the deaths of over 250,000 additional people in the following decades. Even if military complexes are struck with a nuclear weapon, there will be civilian casualties.

### **3. There are environmental concerns which must be considered with a detonation.**

Since 1945, the testing of nuclear weapons has caused more than 2,100 detonations that have happened around the globe. Some have occurred underwater, while others have occurred underground. Every detonation creates radiation that, if someone were exposed to it, could create grave consequences. The threat of multiple simultaneous nuclear detonations could create a prolonged winter that would ravage the planet and potentially cause all life to cease.

### **4. It provides a major terror threat.**

The size of a nuclear weapon is relatively small. With terrorism on the rise around the world, the threat of a “dirty bomb” cannot be ignored. A terrorist group with a nuclear weapon could cause immense and immediate damage that could change the path of an entire society. The Patriot Act of 2001 was in response to airplanes being flown into buildings. Imagine what the aftermath of a nuclear explosion could be.

### **5. Waste from the development of nuclear weapons must be stored somewhere.**

The challenges of storing nuclear waste are evident at the Hanford Site in Washington State. It was the location of the world’s first plutonium production reactor. Up to 9 nuclear reactors and 5 plutonium processors were operating at once, creating 60,000 weapons for the US nuclear arsenal. Now the radioactive waste is stored in 177 storage tanks. Breaches and leaks are still a very real threat, despite the lack of production that occurs there today.

### **6. It is a non-renewable resource.**

Nuclear fuels are plutonium and uranium, which are radioactive metals. This means it is a resource that is non-renewable. Even if the all the nuclear energy from weapons was transitioned into usable energy that wouldn’t produce carbon dioxide, any accident could create a threat to human health that could potentially last for thousands of years. In many ways, nuclear technologies are much like fossil fuels, but with a different threat to the environment.

### **7. A nuclear weapon can fail because of a degraded delivery system.**

Even if the nuclear weapon is viable, the delivery system for the weapon may not be viable. Many nuclear weapons are being maintained with delivery systems that have not been

modernized. In the US, the current stock of ICBM nuclear weapons is expected to remain in service until 2032. Submarine-based nuclear weapons are expected to remain in service until 2042. Unless delivery systems are upgraded and maintained in the same way as their weapon counterparts, they may not be an effective deterrent.

#### **8. They require a skilled workforce to operate.**

Although nuclear weapons can be operated from a distance, a skilled workforce is required to make them a useful tool. Scientists, engineers, and operators are critical to the viability of any nuclear weapon. Without a skilled workforce in support, the safety and security of the nuclear weapons becomes doubtful and that doubt can be enough to make the weapons less of a deterrent.

The pros and cons of nuclear weapons show that they can serve as a deterrent and support “positive” technologies. There is just one question that continues to remain unanswered: what will happen if someone finally calls the bluff and attacks a nation with nuclear capabilities? It is the potential answer to that question which should cause everyone to pause and consider the continued need for these weapons.

### **3. The educational achievements of any person depend largely on family, friends and society.**

**Do you agree or disagree with this statement? Your response should explore both sides of the arguments.**

Educational achievements of an individual largely depends on the personal traits of an individual. The performance of an individual however depend on other external factors such as family, friends and society.

Education is a lasting process. Academic performance in primary education plays a crucial role in obtaining further educational opportunities. Thus, it is necessary to examine how family background affects children's academic achievement at an early stage. There are two pathways through which family influences children's academic performance. Firstly, parents compete for high-quality educational opportunities for their children and better educational opportunities lead to better academic performance. Secondly, parenting behavior and educational support for their children could cultivate children's learning habits and affect academic performance. We also find urban students' academic performance are more heavily affected by their families' socioeconomic status compared with rural students.

Parents and family members play a vital role in the development of an individual. This can be explained using biological and genetic factors. When a learner comes from a family of sharp parents, this trait can be genetically inherited by the student and becomes sharp in school.

Friends have got both direct and indirect influence on the achievement of an individual academically. Friends who positively influence the learner will increase his or her performance while the negative influencer will decrease the performance of the learner

Society plays a very significant role in shaping the behavior of an individual. One develops positive or negative behavior depending upon the kind of exposure one has got in the society around him. This will provide cool environment for learning which will increase the performance of the learner.

It is good to note that the good achievement of an individual depends on him or her. Without parents we would not come in to this world, without friends we not come across enjoyment, fun, hard work, trust, showing good path, helping nature etc. But parents are only main role in the achievement of any person. Without their care, love, willpower no one cannot do anything but I know that few don't have parents and few are handicaps, rumors etc but now-a-days they are also showing their talents hidden in them, they're also achieving great things in their lives.

#### **4. How do you balance between professional and personal life? How do you maintain success? Is success a short term or long term?**

The first step in balancing your work and personal life is recognizing that it is not just going to happen—you have to be proactive and work at it, and the time to do so is now.

Finding a balance between your work and your personal life does not just happen, according to professionals in the field of physician well-being. You have to work at it, and the time to do so is now.

“Balance requires effort, planning, and tradeoffs,” says John Schorling, MD, MPH, internal medicine professor and director of physician wellness programs at the University of Virginia School of Medicine. Referring to a psychology of postponement attributed to physicians, Schorling comments, “We always think life will be better in the next phase—when I’m out of residency, after my fellowship, as soon as my practice is built up.” But, Schorling stresses, life will never change if you are waiting for the future. The first step in finding work-life balance is recognizing that it is not just going to happen—you have to be proactive.

Here are some ways to create a better work-life balance, as well as how to be a supportive manager.

**1. Accept that there is no 'perfect' work-life balance.**

When you hear "work-life balance," you probably imagine having an extremely productive day at work, and leaving early to spend the other half of the day with friends and family. While this may seem ideal, it is not always possible.

Don't strive for the perfect schedule; strive for a realistic one. Some days, you might focus more on work, while other days you might have more time and energy to pursue your hobbies or spend time with your loved ones. Balance is achieved over time, not each day.

**2. Find a job that you love.**

Although work is an expected societal norm, your career shouldn't be restraining. If you hate what you do, you aren't going to be happy, plain and simple. You don't need to love every aspect of your job, but it needs to be exciting enough that you don't dread getting out of bed every morning.

**3. Prioritize your health.**

Your overall physical, emotional and mental health should be your main concern. If you struggle with anxiety or depression and think therapy would benefit you, fit those sessions into your schedule, even if you have to leave work early or ditch your evening spin class. If you are battling a chronic illness, don't be afraid to call in sick on rough days. Overworking yourself prevents you from getting better, possibly causing you to take more days off in the future.

**4. Don't be afraid to unplug.**

Cutting ties with the outside world from time to time allows us to recover from weekly stress, and gives us space for other thoughts and ideas to emerge. Unplugging can mean something simple like practicing yoga instead of checking work emails.

**5. Take a vacation.**

Sometimes, truly unplugging means taking vacation time and shutting work completely off for a while. Whether your vacation consists of a one-day staycation or a two-week trip to Goa, it's important to take time off to physically and mentally recharge.

**6. Make time for yourself and your loved ones.**

While your job is important, it shouldn't be your entire life. You were an individual before taking this position, and you should prioritize the activities or hobbies that make you happy

When planning time with your loved ones, create a calendar for romantic and family dates. It may seem weird to plan one-on-one time with someone you live with, but it will ensure that you spend quality time with them without work-life conflict. Just because work keeps you busy doesn't mean you should neglect personal relationships.

**7. Set boundaries and work hours.**

Set boundaries for yourself and your colleagues, to avoid burnout. When you leave the office, avoid thinking about upcoming projects or answering company emails. Consider having a separate computer or phone for work, so you can shut it off when you clock out. If that isn't possible, use separate browsers, emails or filters for your work and personal platforms.

**8. Set goals and priorities (and stick to them).**

Pay attention to when you are most productive at work and block that time off for your most important work-related activities. Avoid checking your emails and phone every few minutes, as those are major time-wasting tasks that derail your attention and productivity. Structuring your day can increase productivity at work, which can result in more free time to relax outside of work.

Work-life balance will mean different things to different people because, after all, we all have different life commitments. In our always-on world, balance is a very personal thing, and only you can decide the lifestyle that suits you best.

## **5. Is Life better In a small town or a big city?**

It is known that what is good for one person may not be good for another person. For example, some people always prefer to live in a big city, and some people prefer to live in a small town. Everyone has different points of view for that. A person has different thinking and reason to choose a place for living. Jobs, education, family, lifestyle, personal development and some other factors are responsible for choosing their place to live. Both living in a big city and living in small town has its advantages and disadvantages. First of all, In a small town, you can build up a good community.

Education is one of the reasons for living in a big town or city. Both cities and small towns have good and bad schools. However, there are a lot of options in a big city. Most small towns have one or two high schools. On the other hand, a big city has more schools than that. In a big city, you have the option for your career making field. A Big city has universities, colleges and schools. Because of that you can go to more sporting venues, like NBA Basketball games, NBA Football games, as well as many minor league or major league sporting events. Small towns offer you local high school sports. When you live in a big city, you have much greater access to a variety of entertainment venues, and you are exposed to all sorts of different cultures. Cities also have all

sorts of music festivals, comedy festivals, and other events that happen on a fairly regular basis. By contrast, small towns generally don't have nearly as much variety. Furthermore, Small towns rarely have a formal public transportation system. People must need to buy their own vehicle or depend on others for a ride.

Even though I have never lived in a city, I think life is better in a small town because the community is close knit. Small towns have less crime and less traffic as well.

In a small town everybody seems to know each other. Strangers smile at you when you pass them in public. It also seems like people in small towns are more willing to reach out and help you when you are in need. Cities seem to have a lot of crime and in small towns crime is very low. You don't have to worry about locking doors or closing windows and you are less likely to get mugged or robbed. Children are also safe walking to school alone or playing outside with little or no supervision.

Traffic is another great thing about living in a small town. It takes less time to drive from one end of town to the other than it does in the city. You also don't have to wait very long at intersections or spend much time searching for a free parking spot.

## **6. Culture and traditions are as important as science to the progress and well being of the society**

Traditions represent a critical piece of our culture. They help form the structure and foundation of our families and our society. They remind us that we are part of a history that defines our past, shapes who we are today and who we are likely to become. Once we ignore the meaning of our traditions, we're in danger of damaging the underpinning of our identity.

- Tradition contributes a sense of comfort and belonging. It brings families together and enables people to reconnect with friends.
- Tradition reinforces values such as freedom, faith, integrity, a good education, personal responsibility, a strong work ethic, and the value of being selfless.
- Tradition provides a forum to showcase role models and celebrate the things that really matter in life.
- Tradition offers a chance to say "thank you" for the contribution that someone has made.
- Tradition enables us to showcase the principles of our Founding Fathers, celebrate diversity, and unite as a country.
- Tradition serves as an avenue for creating lasting memories for our families and friends.
- Tradition offers an excellent context for meaningful pause and reflection.

As leaders, role models, and parents, we must strive to utilize every opportunity available to us to reinforce the values and beliefs that we hold dear. The alternative to action is taking these values for granted. The result is that our beliefs will get so diluted, over time, that our way of life will become foreign to us. It's like good health. You may take it for granted until you lose it. If we disregard our values, we'll open our eyes one day and won't be able to recognize "our world" anymore. The values that support the backbone of our country, our family, and our faith will have drifted for so long that the fabric of our society will be torn.

Teach your generations about your roots, culture, traditions and customs however make them understand to follow only those which are practically feasible.

I think science is a part of culture. There is no sense facing them each other. They require each other: without culture science cannot exist and at a certain point of cultural evolution science appears and develops parallel with culture. Both are manifestation of human brain, creativity and human relationship to the environment as well as the universe of facts and ideas.

Scientists are people. People have culture.

Therefore, when making scientific assessments or in formulating scientific hypothesis people's culture will indirectly make influences; life outside of the lab, influences life in the lab.

Additionally, where scientist do research, the extent, and in what fields may be influenced by culture.

A perfect example is Ramanujan, mathematician, who said famously "none of his math conjectures meant anything unless there where signs from god."

## **7. The tight curriculum of our education system leaves no room for imagination and creativity.**

**Write a response that expresses your thoughts on this statement. To what extent do you agree or disagree? Explain your reasoning.**

I wholeheartedly agree that the tight curriculum of our current education system leaves no room for imagination and creativity. The academic pressure is so much on students, they have forgotten their innate creativity and imagination. They are always under the pressure of studies. This pressure is put on them from the beginning of the session. Formative assessment activities of all the subjects, studies and their assignments of all the subjects, unit tests, co-curricular activities, homework, classwork, projects, practical exams, summative examination,

date sheet, result, again the new term, etc. So much of pressure on the young growing children! Are we educating them or training them in the art of being stressful throughout their childhood?

Even while studying inside the class, they feel so burdened; they have come to dislike education. as a result, there is hardly any creativity left in our country. Every year all the illustrious prizes for creativity are won by non-Indians or Indians living in advanced countries. Our educationists must rethink and revise the curriculum to encourage creativity and imagination in children.

With each passing day, education is becoming more difficult and the system is getting stricter. With such heavy rules and regulations in academics, students hardly have the opportunity to dream and imagine. Children are known for their imagination and it is due to imagination that they can explore new creative arenas. This is creating extremely methodical individuals but the true meaning of education is somewhat missing. Education teaches us to be more vocal about our likings and disliking and improves our imaginative skill and creativity.

## **8. The importance of weaponry.. How it affects the current world**

Weapons are used to increase the efficacy and efficiency of activities such as hunting, crime, law enforcement, self-defense, and warfare. In broader context, weapons may be construed to include anything used to gain a tactical, strategic, material or mental advantage over an adversary or enemy target.

The time is right for a fundamental rethinking of the role of nuclear weapons in national defense and of the composition of our nuclear forces. The Cold War is over, but it has been replaced by new threats to our national security. Technology, here and abroad, is inexorably advancing, creating both dangers and opportunities for the United States. This paper analyzes the future role of nuclear weapons in national security, describes the roles and limitations of advanced conventional weapons in meeting strategic needs, and suggests several alternate scenarios for future U.S. nuclear forces.

The principal role of nuclear weapons is to deter potential adversaries from an attack on the United States, our allies, or our vital interests. Russia maintains very large strategic and tactical nuclear forces. China is actively modernizing its nuclear arsenal. India and Pakistan have dramatically demonstrated the ability of midlevel technology states to develop or acquire nuclear weapons. There are grave concerns about the future proliferation of nuclear weapons among such countries as North Korea, Iraq, and Iran. The nuclear age is far from over.

Advances in conventional weapons technology suggest that by 2020 precision long-range conventional weapons may be capable of performing some of the missions currently assigned to nuclear weapons. Today, uncertainty in the location of road mobile missiles carrying weapons of mass destruction might require a nuclear weapon for assured destruction. Future real-time imagery and battle management, combined with precision strike long-range missiles, may mean that a conventional weapon could effectively destroy such targets.

Some targets require the energy of a nuclear weapon for their destruction. However, precision targeting can greatly reduce the nuclear yield required to destroy such targets. Only a relatively few targets require high nuclear yields. Advantages of lower yields include reduced collateral damage, arms control advantages to the United States, and the possibility that such weapons could be maintained with higher confidence and at lower cost than our current nuclear arsenal.

Now is the time to reexamine the role and composition of our future nuclear forces. New technologies take at least a decade to move from the concept stage to the point where we can rely on them for our nations defense. And, advance planning is already under way for the replacements of our nuclear capable missiles, aircraft, and sub-marines. Prudent thought given to this crucial subject will reap great dividends for the United States and for peace in the world.

Q) 1

1 1

1 1 1

```
void patternPrint(int num)
```

```
{
```

```
    int print=1,i,j;
```

```
    for(i=0;i<num;i++)
```

```
{
```

```
    for(j=0;j<=i;j++);
```

```
{
```

```
        printf("%d ",print);
```

```
}
```

```
    printf("\n");
```

```
}
```

Ans: semicolon after for loop

```
void patternPrint(int num){
```

```
    int print=1,i,j;
```

```
    for(i=0;i<num;i++) {
```

```
        for(j=0;j<=i;j++)
```

```
{
```

```
        printf("%d ",print);
```

```
}
```

```
        printf("\n");
```

```
}
```

```
}
```

Q) return true if given date is july 5<sup>th</sup>.

```
int checkBirthDay(char* month, int day)
```

```
{
```

```
    if((strcmp(month, "July")) || (day == 5))
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

Ans: both conditions has to be satisfied. so and operator has to be applied.

strcmp return 0 when equal. so check 0 or not

```
int checkBirthDay(char* month, int day)
```

```
{
```

```
    if((strcmp(month, "July")==0) && (day == 5))
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

Q) find x power y, given function that computes positive power only.

```
float allExponent(int baseValue,int exponentValue){  
    float res=1;  
    if(exponentValue >=0)  
    {  
        res=(float) positiveExponent(baseValue,exponentValue);  
    }  
    else  
    {  
        //write your code here for negative value of exponent  
    }  
    return res;  
}
```

Ans:  $1/x$  equals to  $x^{-1}$ .

```
float allExponent(int baseValue,int exponentValue)
{
    float res=1;
    if(exponentValue >=0)
    {
        res=(float) positiveExponent(baseValue,exponentValue);
    }
    else
    {
        res=(float) positiveExponent(baseValue,-exponentValue);
        res=1/res;
    }
    return res;
}
```

Q) check whether given points form a triangle or not

```
int isTriangle(Point *P1, Point *P2, Point *P3)
{
    //write your code here
}
```

Ans: area of a triangle given three points is

$$(1/2) * (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)).$$

if this is zero then it does not form a triangle

```
int isTriangle(Point *P1, Point *P2, Point *P3)
```

```
{
    int a = P1->x * (P2->y - P3->y) + P2->x * (P3->y - P1->y) +
           P3->x * (P1->y - P2->y);

    if (a == 0)
        return 0;
    else
        return 1;
}
```

Q) program that computes sum of all elements in a matrix

```
int matrixSum(int rows, int columns, int **matrix)
{
    int i,j,sum=0;
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
            sum+=matrix(i)(j);
    }
    return sum;
}
```

Ans: for indexing square brackets has to be used not parenthesis.

```
int matrixSum(int rows, int columns, int **matrix)
{
    int i,j,sum=0;
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
            sum+=matrix[i][j];
    }
    return sum;
}
```

Q) sorting problem

```
void selectionSortArray(int len,int *arr)
{
    int x=0,y=0;
    for(x=0;x<len;x++){
        int index_of_min=x;
        for(y=x;y<len;y++){
            if(arr[index_of_min]>arr[y]){
                index_of_min=y;
            }
        }
        int temp=arr[x];
        arr[x]=arr[index_of_min];
        arr[index_of_min]=temp;
    }
}
```

Ans: index\_of\_min element should be compared with y not with x

```
void selectionSortArray(int len,int *arr)
{
    int x=0,y=0;
    for(x=0;x<len;x++){
        int index_of_min=x;
        for(y=x;y<len;y++){
            if(arr[index_of_min]>arr[y]){
                index_of_min=y;
            }
        }
        int temp=arr[x];
        arr[x]=arr[index_of_min];
        arr[index_of_min]=temp;
    }
}
```

Q) program to perform descending order sorting

```
void descendingSortArray(int len,int *arr)
```

```
{
```

```
    int small,pos,i,j,temp;
```

```
    for(i=0;i<=len-1;i++){
```

```
        for(j=i;j<len;j++){
```

```
            temp=0;
```

```
            if(arr[i]>arr[j]){
```

```
                temp=arr[i];
```

```
                arr[i]=arr[j];
```

```
                arr[j]=temp;
```

```
        }
```

```
    }
```

```
}
```

```
}
```

Ans: given program performs ascending order sorting.

reverse the condition to get descending order.

```
void descendingSortArray(int len,int *arr)
{
    int small,pos,i,j,temp;
    for(i=0;i<=len-1;i++){
        for(j=i;j<len;j++){
            temp=0;
            if(arr[i]<arr[j]){
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```

Q) replace all odd elements with minimum element and  
all even elements with maximum element

```
void replaceMinMax(int size, int *arr)
{
    int i;
    if(size>0) {
        int max=arr[0];
        int min=arr[0];
        for(i=0;i<size;i++){
            if(max<arr[i]){
                max=arr[i];
            }
            else if(min>arr[i]){
                min=arr[i];
            }
        }
        for(i=0;i<size;i++){
            if(arr[i]%2==0)
                arr[i]=max;
            else
                arr[i]=min;
        }
    }
}
```

Ans: array element access is done through square brackets

not with '.' operator

```
void replaceMinMax(int size, int *arr){  
    int i;  
    if(size>0){  
        int max=arr[0];  
        int min=arr[0];  
        for(i=0;i<size;i++){  
            if(max<arr[i]){  
                max=arr[i];  
            }  
            else if(min>arr[i]){  
                min=arr[i];  
            }  
        }  
        for(i=0;i<size;i++){  
            if(arr[i]%2==0)  
                arr[i]=max;  
            else  
                arr[i]=min;  
        }  
    }  
}
```

Q) 1 1

2 2 2 2

3 3 3 3 3 3

```
void printPattern(int num)
```

```
{
```

```
    int i,j;
```

```
    for(i=1;i<=num;i++)
```

```
{
```

```
        for(j=i;j<=2*i;j++)
```

```
{
```

```
            printf("%d",j);
```

```
}
```

```
        printf("\n");
```

```
}
```

```
}
```

Ans: every time j starts from 1 not from i.

if j is printed then it will be 1 2, 1 2 3 4, 1 2 3 4 5 6 like that.

to get 1 1, 2 2 2 2 print i

```
void printPattern(int num)
{
    int i,j;
    for(i=1;i<=num;i++)
    {
        for(j=1;j<=2*i;j++)
        {
            printf("%d",i);
        }
        printf("\n");
    }
}
```

Q) print the Fibonacci series with given number of values

```
void printFibonacci(int num)
```

```
{
```

```
    int i;
```

```
    long sum=0;
```

```
    long num1=0;
```

```
    long num2=1;
```

```
    for(i=1;i<num;++i)
```

```
{
```

```
    print("%d ",num1);
```

```
    sum=num1+num2;
```

```
    num2=sum;
```

```
    num1=num2;
```

```
}
```

```
}
```

Ans: as num values has to be printed and initially i starts with 1,

condition must have '=' or start i with 0

format specifier for long int is %ld

```
void printFibonacci(int num){  
    int i;  
    long sum=0;  
    long num1=0;  
    long num2=1;  
    for(i=1;i<=num;++i) {  
        printf("%ld ",num1);  
        sum=num1+num2;  
        num2=sum;  
        num1=num2;  
    }  
}
```

(or)

```
void printFibonacci(int num){  
    int i;  
    long sum=0;  
    long num1=0;  
    long num2=1;  
    for(i=0;i<num;++i){  
        printf("%ld ",num1);  
        sum=num1+num2;  
        num2=sum;  
        num1=num2;  
    }  
}
```

Q) count number of adjacent values are equal upto that position.

For first position previous element is assumed to be 0.

Example1:

Input: 6, [1, 1, 0, 0, 1, 0]

Output: 0 1 0 2 0 0

Example2:

Input: 8, [0, 0, 0, 1, 0, 1, 1, 1]

Output: 1 2 3 0 0 0 4 5

```
void manchester(int size,int* arr){  
    bool result;  
    int* res=(int*)malloc(sizeof(int)*size);  
    int count=0;  
    for(int i=0;i<size;i++){  
        if(i==0)  
            result=(arr[i]==0);  
        else  
            result=(arr[i]==arr[i-1]);  
        res[i]=(result)?(0):(++count);  
    }  
    for(int i=0;i<size;i++){  
        printf("%d ",res[i]);  
    }  
}
```

Ans: in ternary operation for condition only result is used,  
if result is zero then it will become false and count is incremented.  
but count has to be incremented when result is 1.  
so change it as result==0  
data type bool is not supported in c. change it to int.

```
void manchester(int size,int* arr)
{
    int result;
    int* res=(int*)malloc(sizeof(int)*size);
    int count=0;
    for(int i=0;i<size;i++)
    {
        if(i==0)
            result=(arr[i]==0);
        else
            result=(arr[i]==arr[i-1]);
        res[i]=(result==0)?(0):(++count);
    }
    for(int i=0;i<size;i++)
    {
        printf("%d ",res[i]);
    }
}
```

Q) reverse the second half of array elements

Example1:

Input: 7, [1, 2, 3, 4, 5, 6, 7]

Output:1, 2, 3, 7, 6, 5, 4

Example2:

Input: 4, [1, 2, 3, 4]

Output:1, 2, 4, 3

```
void reverseHalfArray(int size,int *inputList)
```

```
{
```

```
int i,temp;
```

```
for(i=0;i<size/2;i++)
```

```
{
```

```
    temp=inputList[size-1];
```

```
    inputList[size-1]=inputList[i];
```

```
    inputList[i]=temp;
```

```
    size-=1;
```

```
}
```

```
}
```

Ans: middle element lies at  $\text{size}/2$  position. so start  $i$  from  $\text{size}/2$ .  
as size is decreasing after every swap, stop when  $i$  reaches size.

```
void reverseHalfArray(int size,int *inputList)
{
    int i,temp;
    for(i=size/2;i<size;i++)
    {
        temp=inputList[size-1];
        inputList[size-1]=inputList[i];
        inputList[i]=temp;
        size-=1;
    }
}
```

Q) a

a b  
a b c  
a b c d  
a b c d e

```
void printCharacterPattern(int num)
```

```
{
```

```
    int i,j;  
    char ch='a';  
    char print;  
    for(i=0;i<num;i++){  
        print=ch;  
        for(j=0;j<=i;j++)  
            printf("%c",ch++);  
        printf("\n");  
    }  
}
```

Ans: if we are updating ch, then we will get continuation alphabet from the previous line.  
so update and print the value from variable print.

```
void printCharacterPattern(int num)
{
    int i,j;
    char ch='a';
    char print;
    for(i=0;i<num;i++){
        print=ch;
        for(j=0;j<=i;j++)
            printf("%c",print++);
        printf("\n");
    }
}
```

Q) give remainder when given number is divided by number of digits in it.

Example:

Input: 125

Output: 2 (125 mod 3)

```
int countDigits(int num)
```

```
{
```

```
    int count = 0;
```

```
    while(num != 0)
```

```
{
```

```
        num = num/10;
```

```
        count ++;
```

```
}
```

```
    return (num % count);
```

```
}
```

ans: num becomes 0 at the end of process. so save it in temp

```
int countDigits(int num)
```

```
{
```

```
    int count = 0;
```

```
    int t=num;
```

```
    while(num != 0)
```

```
{
```

```
        num = num/10;
```

```
        count ++;
```

```
}
```

```
    return (t % count);
```

```
}
```

Q) find the number of days between given two dates.

Difference shall be always either positive or zero.

Example1:

Input: {day:02, month:05, year:2013}

{day:02, month:06, year:2013}

Output: 31

Example2:

Input: {day:01, month:06, year:2011}

{day:01, month:06, year:2012}

Output: 366

```
int difference_in_dates(Date *date1, Date *date2)
{
    // write your code here
}
```

ans: Let the given two dates be "1-Feb-2000" and "1-Feb-2004"

dt1 = {1, 2, 2000};

dt2 = {1, 2, 2004};

Count number of days before date1. Let this count be n1.

Every leap year adds one extra day (29 Feb) to total days.

$n1 = 2000 * 365 + 31 + 1 + \text{Number of leap years}$

Count of leap years for a date 'd/m/y' can be calculated using following formula:

Number leap years

=  $\text{floor}(y/4) - \text{floor}(y/100) + \text{floor}(y/400)$  if  $m > 2$

=  $\text{floor}((y-1)/4) - \text{floor}((y-1)/100) + \text{floor}((y-1)/400)$  if  $m \leq 2$

All above divisions must be done using integer arithmetic so that the remainder is ignored.

For 01/01/2000, leap year count is  $1999/4 - 1999/100 + 1999/400$  which is  $499 - 19 + 4 = 484$

Therefore n1 is  $2000 * 365 + 31 + 1 + 484$

Similarly, count number of days before date2. Let this count be n2.

Finally return  $n2 - n1$

```
int difference_in_dates(Date *date1, Date *date2)
{
    long int n1, n2;
    int i, y;
    int monthDays[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    n1 = date1->year * 365 + date1->day;
    for (i=0; i<date1->month - 1; i++)
        n1 += monthDays[i];
    y = date1->year;
    if (date1->month <= 2)
        y--;
    n1 += y/4 - y/100 + y/400;

    n2 = date2->year * 365 + date2->day;
    for (i=0; i<date2->month - 1; i++)
        n2 += monthDays[i];
    y = date2->year;
    if (date2->month <= 2)
        y--;
    n2 += y/4 - y/100 + y/400;

    return (n2 - n1);
}
```

Q) reverse the elements of an array.

Example:

Input: 4, [1, 2, 3, 4]

Output: 4 3 2 1

```
int arrayReverse(int len,int* arr)
```

```
{
```

```
    int i,temp,originallen = len;
```

```
    for(i= 0;i<=originallen/2;i++)
```

```
{
```

```
    temp=arr[len-1];
```

```
    arr[len-1] =arr[i];
```

```
    arr[i] = temp;
```

```
    len -= 1;
```

```
}
```

```
}
```

Ans:  $\text{len}/2$  number of swappings are enough.

but in code iterations starts from 0 and '=' is included in condition.

so iterates one time extra.

```
int arrayReverse(int len,int* arr)
{
    int i,temp,originallen = len;
    for(i= 0;i<originallen/2;i++)
    {
        temp=arr[len-1];
        arr[len-1] =arr[i];
        arr[i] = temp;
        len -= 1;
    }
}
```

Q) count number of elements greater than twice the given element in an array

Example1:

Input: 5, 3, [0, -3, 1, -6, 2]

Output: 0

Example2:

Input: 5, 3, [11, 2, 3, 14, 5]

Output: 2

```
int countElement(int size,int numK,int *inputList)
{
    int i,cou-nt = 0;
    for(i=0;i<size;i++)
    {
        if(inputList[i]>2numK)
            cou-nt+=1;
    }
    return cou-nt;
}
```

Ans:

2numK - there should be explicit multiplication operator like  $2 * \text{numK}$   
cou-nt - is not allowed in variable name.

```
int countElement(int size,int numK,int *inputList)
```

```
{
```

```
    int i,count = 0;
```

```
    for(i=0;i<size;i++)
```

```
{
```

```
        if(inputList[i]>2*numK)
```

```
            count+=1;
```

```
}
```

```
    return count;
```

```
}
```

Q) remove an element at a given index from an array

Example1:

Input: 9, 3, [1, 2, 3, 4, 5, 6, 7, 8, 9]

Output: 1 2 3 5 6 7 8 9

Example2:

Input: 6, 6, [1, 2, 3, 4, 5, 6]

Output: 1 2 3 4 5 6

```
void removeElement(int size,int indexValue,int *inputList){  
    int i,j;  
    if(indexValue<size) {  
        for(i= indexValue;i<size-1;i++) {  
            inputList[i] = inputList[i++];  
        }  
        for(i=0;i<size-1;i++)  
            printf("%d ",inputList[i]);  
    }  
    else {  
        for(i=0;i<size;i++)  
            printf("%d ",inputList[i]);  
    }  
}
```

Ans: every element after the given index should be moved one position left. Element at  $i+1$  is stored at  $i$ .  
in given code  $i++$  is used instead of  $i+1$

```
void removeElement(int size,int indexValue,int *inputList)
{
    int i,j;
    if(indexValue<size)
    {
        for(i= indexValue;i<size-1;i++)
        {
            inputList[i] = inputList[i+1];
        }
        for(i=0;i<size-1;i++)
        {
            printf("%d ",inputList[i]);
        }
    }
    else
    {
        for(i=0;i<size;i++)
        {
            printf("%d ",inputList[i]);
        }
    }
}
```

Q) find the largest element among two input lists,  
given a function to sort the list of elements

Example:

Input: 5, [1, 5, 2, 7, 9]

4, [10, 4, 6, 3]

Output:10

```
int *sortArray(int len,int* arr){
```

```
    int i=0,j=0,temp =0;
```

```
    for(i=0;i<len;i++) {
```

```
        for(j=i+1;j<len;j++) {
```

```
            if(arr[i]>arr[j]) {
```

```
                temp = arr[i];
```

```
                arr[i]=arr[j];
```

```
                arr[j]=temp;
```

```
            }
```

```
        }
```

```
        return arr;
```

```
}
```

```
int findMaxElement(int len1,int* arr1,int len2,int* arr2)
```

```
{
```

```
    //write your code
```

```
}
```

Ans: sort both arrays and find the largest among the last element of both arrays

```
int findMaxElement(int len1,int* arr1,int len2,int* arr2)
{
    arr1=sortArray(len1,arr1);
    arr2=sortArray(len2,arr2);
    if(arr1[len1-1]>arr2[len2-1])
        return arr1[len1-1];
    else
        return arr2[len2-1];
}
```

Q) print 0 if current bit equals to previous bit, otherwise 0.

For first bit assume previous bit as 0

Example:

Input: 8, [0, 1, 0, 0, 1, 1, 1, 0]

Output: 0 1 1 0 1 0 0 1

```
void manchester(int len,int* arr)
{
    int* res =(int*)malloc(sizeof(int)*len);
    res[0] = arr[0];
    for(int i=1;i<len;i++){
        res[i] = (arr[i]==arr[i-1]);
    }
    for(int i=0;i<len;i++)
        printf("%d ",res[i]);
}
```

Ans: if equal comparison returns 1, but as per question we need 0 in when equal. so apply not operator

```
void manchester(int len,int* arr)
{
    int* res =(int*)malloc(sizeof(int)*len);
    res[0] = arr[0];
    for(int i=1;i<len;i++){
        res[i] = !(arr[i]==arr[i-1]); // or res[i] =(arr[i]!=arr[i-1]);
    }
    for(int i=0;i<len;i++)
        printf("%d ",res[i]);
}
```

Q) find product of maximum two numbers out of three numbers

Example:

Input: 5, 2, 7

Output:35

```
int multiplyNumber(int numA,int numB,int numC)
{
    int result,min,max,mid;
    max = (numA>numB) ?numA>numC) ?numA:numC) :
        (numB>numC) ? numB:numC);
    min = (numA<numB) ? numA<numC) ? numA:numC) :
        (numB<numC)?numB:numC);
    mid = (numA+numB+numC)-(min+max);
    result = (max*int mid);
    return result;
}
```

Ans: parenthesis not properly opened and closed.

int has to be removed during multiplication

```
int multiplyNumber(int numA,int numB,int numC)
{
    int result,min,max,mid;
    max = (numA>numB) ? (numA>numC ? numA : numC) :
        (numB>numC)?numB:numC;
    min = (numA<numB) ? (numA<numC ? numA:numC) :
        (numB<numC)?numB:numC;
    mid = (numA+numB+numC)-(min+max);
    result = (max*mid);
    return result;
}
```

Q)

```
float median(int size,int * inputList){  
    int start_index = 0;  
    int end_index = size-1;  
    float res = -1;  
    if(size%2!=0)  
    {  
        int median_order = ((size+1)/2);  
        res = (float)quick_select(inputList , start_index, end_index,  
                                median_order);  
    }  
    else  
    {  
        // write your code here  
    }  
    return res;  
}
```

Ans: for even size median is average of middle two numbers

```
float median(int size,int * inputList){  
    int start_index = 0;  
    int end_index = size-1;  
    float res = -1;  
    if(size%2!=0)  
    {  
        int median_order = ((size+1)/2);  
        res = (float)quick_select(inputList , start_index, end_index,  
                                median_order);  
    }  
    else  
    {  
        int median_order = ((size)/2);  
        float res1;  
        res = (float)quick_select(inputList , start_index, end_index,  
                                median_order);  
        res1=(float)quick_select(inputList , start_index, end_index,  
                                median_order+1);  
        res=(res+res1)/2;  
    }  
    return res;  
}
```

Q) given three point find whether they forms a right angled triangle or not.

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
}
```

Ans: find the distance between three points. let them be a,b,c.  
if  $a^2+b^2=c^2$  then it will be right triangle

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
    double d1,d2,d3,max,min,mid;
    d1=point_calculateDistance(p1,p2);
    d2=point_calculateDistance(p2,p3);
    d3=point_calculateDistance(p1,p3);

    max = (d1>d2)?(d1>d3?d1:d3):(d2>d3)?d2:d3;
    min = (d1<d2)?(d1<d2?d1:d3):(d2<d3)?d2:d3;
    mid = (d1+d2+d3)-(min+max);

    if(max*max==min*min+mid*mid)
        return 1;
    else
        return 0;
}
```

Q) find number of occurrences of a given element.

```
int countOccurance(int len,int value,int *arr)
{
    int i=0,count =0;
    while(i<len){
        if(arr[i] == value)
            count += 1;
    }
    return count;
}
```

Ans: updation of index variable is missed.

```
int countOccurance(int len,int value,int *arr)
{
    int i=0,count =0;
    while(i<len){
        if(arr[i] == value)
            count += 1;
        i+=1;
    }
    return count;
}
```

Q) 1 1

1 1 1 1

1 1 1 1 1 1

```
void drawPrintPattern(int num)
```

```
{
```

```
    int i,j,print = 1;
```

```
    for(i =1;i<=num;i++)
```

```
{
```

```
        for(j=1;j<=2*i;j++);
```

```
{
```

```
        printf("%d ",print);
```

```
}
```

```
        printf("\n");
```

```
}
```

```
}
```

Ans: semicolon at the end of for loop.

```
void drawPrintPattern(int num)
{
    int i,j,print = 1;
    for(i =1;i<=num;i++)
    {
        for(j=1;j<=2*i;j++)
        {
            printf("%d ",print);
        }
        printf("\n");
    }
}
```

Q) count occurrences of a even numbers that appears adjacently.

Example:

Input: 8, [1,2,2,2,5,5,4,4]

Output:3

```
int sameElementCount(int size,int *inputList)
{
    int i,count =0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i] == inputList[i+1]))
            count++;
    }
    return count;
}
```

Ans: i+1 instead of i++ in indexing

```
int sameElementCount(int size,int *inputList)
{
    int i,count =0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i] == inputList[i+1]))
            count++;
    }
    return count;
}
```

Q) Input: 5

Expected output:

a  
ab  
abc  
abcd  
abcde

```
#include<stdio.h>
void printcharacterpattern( int num)
{
    int i,j,value=1;
    char ch = 'a';
    char print = ch;
    for(int i=1;i<=num;i++,printf("\n"))
    {
        ch=print;
        for(j=1;j<=i;j++)
            printf("%c",ch++);
    }
}
int main()
{
    int num;
    scanf("%d",&num);
    printcharacterpattern(num);
}
```

Q) print the color given a number

```
#include<stdio.h>

void printcolor(int num){

    switch (num)  {

        case 1:
            printf("red");
        case 2:
            printf("black");
        case 3:
            printf("white");
        case 4:
            printf("green");
        default:
            printf("no color");
            break;
    }
}
```

```
int main()
```

```
{
    printcolor(3);
    return 0;
}
```

Ans: break has to be used along with each case

```
#include<stdio.h>

void printcolor(int num)
{
    switch (num)
    {
        case 1:
            printf("red"); break;
        case 2:
            printf("black"); break;
        case 3:
            printf("white"); break;
        case 4:
            printf("green"); break;
        default:
            printf("no color");
            break;
    }
}

int main()
{
    printcolor(3);
    return 0;
}
```

Q) 1

1 1

1 1 1

```
#include <stdio.h>
int printpattern(int n)
{
    int i,j,print =1;
    for(i=1;i<=n;i++)
        for(j=1;j<=2 * i;j++)
        {
            printf("%d",print );
        }
    printf("\n");
}
```

```
int main(void)
{
    printpattern(5);
    return 0;
}
```

Ans: i loop has j loop and new line printing as its body.

So enclose them in braces

```
int printpattern(int n)
{
    int i,j,print =1;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=2 * i;j++)
        {
            printf("%d",print );
        }
        printf("\n");
    }
}
```

```
int main(void)
{
    printpattern(5);
    return 0;
}
```

Q) reverse the elements of given array

```
#include <stdio.h>

int arrayReverse(int *arr,int len)

{

int i,temp,originallen=len;
for(i=0;i<=originallen/2;i++)

{

temp=arr[len-1];
arr[len-1]=arr[i];
arr[i]=temp;

len+=1;

}

return arr;

}
```

```
int main(void) {

int a[]={1,2,3,4,5,6},*b,i;
b=arrayReverse(a,6);
for(i=0;i<6;i++)
printf("%d ",b[i]);
return 0;

}
```

Ans: len has to move towards right. So decrement it.

No of swapping should be size/2. So remove '=' in condition  
as started from 0

```
#include <stdio.h>

int arrayReverse(int *arr,int len)

{
int i,temp,originallen=len;
for(i=0;i<originallen/2;i++)
{
temp=arr[len-1];
arr[len-1]=arr[i];
arr[i]=temp;
len-=1;
}
return arr;
}

int main(void) {
int a[]={1,2,3,4,5,6},*b,i;
b=arrayReverse(a,6);
for(i=0;i<6;i++)
printf("%d ",b[i]);
return 0;
}
```

Q) print the grade given score

```
#include<stdio.h>

char checkGrade(int score)

{
    if(score<=60)
        return 'D';
    else if((61>=score)&&(score<=75))
        return 'C';
    else if((76>=score)&&(score<=90))
        return 'B';
    else
        return 'A';
}

int main()

{
    int score;
    scanf("%d",&score);
    printf("%c", checkGrade(score));
    return 0;
}
```

Ans: range condition should be  $\leq$  and  $\geq$  combination

```
#include<stdio.h>

char checkGrade(int score)
{
    if(score<=60)
        return 'D';
    else if((61<=score)&&(score<=75))
        return 'C';
    else if((76<=score)&&(score<=90))
        return 'B';
    else
        return 'A';
}

int main()
{
    int score;
    scanf("%d",&score);
    printf("%c", checkGrade(score));
    return 0;
}
```

Q) find difference in seconds between two times

TESTCASE

TestCase1:

Input:

Time:1:58:42, Time:2:1:45

Expected Return values:

183

Testcase 2

Input:

Time:3:49:57, Time:2:45:57

Expected Return Values

3600

```
int diff_in_times(TIME *t1, TIME *t2){  
    // write your code here  
}
```

Ans: convert time into seconds and then find difference.

If difference is negative change it to positive

H:M:S -> H\*3600 + M\*60 + S

```
#include<stdio.h>

typedef struct {
    int h,m,s;
}TIME;

int toSeconds(TIME * gt){
    int in_seconds;
    in_seconds = gt->h * 3600 + gt->m * 60 + gt->s;
    return in_seconds;
}

int diff_in_times(TIME *t1, TIME *t2){
    int t5,t6,res;
    t5= toSeconds(t1);
    t6= toSeconds(t2);
    res= t5-t6;
    if (res<0)
        res=-res;
    return res;
}

int main()
{
    TIME t1 = {1,58,42}, t2 = {1,59,48};
    printf("%d", diff_in_times(&t1, &t2));
    return 0;
}
```

Q) 1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

```
void printPattern(int n)
```

```
{
```

```
// write your code here
```

```
}
```

```
#include <stdio.h>
```

```
void printPattern(int n){
```

```
int i,j;
```

```
for(i=1;i<=n;i++) {
```

```
    for(j=1;j<=i;j++) {
```

```
        printf("%d",j);
```

```
    }
```

```
    for(j=i-1;j>=1;j--) {
```

```
        printf("%d",j);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
}
```

(Or)

```
void printPattern(int n)
{
    int i,j;
    long int num=0;
    for(i=0;i<n;i++)
    {
        num=num*10+1;
        printf("%ld\n", num*num);
    }
}
```

```
int main()
```

```
{
    printPattern(8);
    return 0;
}
```

Q) The method `deleteDuplicate(int len, int arr[])` takes an array as an input. it is supposed to remove duplicate integers from the input array arr such that for each distinct integer the first occurrence is retained and all the duplicates elements following it are removed

Example:

Input: (2,3,2,2,5,6,6,7)

Output: 2, 3, 5, 6, 7

```
int* deleteDuplicate (int *arr, int len)
```

```
{
```

```
    int i,j,k=0;
```

```
    for(i=0;i<len;i++)
```

```
{
```

```
    for(j=i+1;j<len;j++)
```

```
{
```

```
        if(arr[j]==arr[i])
```

```
{
```

```
            arr[k]=arr[k+1];
```

```
}
```

```
        len=len-1;
```

```
        count=count+1;
```

```
        j=i;
```

```
}
```

```
}
```

```
    return arr;
```

```
}
```

Ans: when a match occurs not a single element all the following elements has to be shifted right one position.

Count has to be increased, size has to be decreased, j has to be updated to j-1 when only match occurs

```
#include <stdio.h>

int count=0;

int* deleteDuplicate (int *arr, int len)

{
    int i,j,k=0;
    for(i=0;i<len;i++)
    {
        for(j=i+1;j<len;j++)
        {
            if(arr[j]==arr[i])
            {
                for(k=j;k<len;k++)
                    arr[k]=arr[k+1];
                len=len-1;
                count=count+1;
                j--;
            }
        }
    }
    return arr;
}
```

```
int main(void)
{
int a[]={2,3,2,2,2,6,6,7},i,*b;
b=deleteDuplicate(a,8);
for(i=0;i<8-count;i++)
printf("%d ",b[i]);
return 0;
}
```

Q) Given a string str, write a program to eliminate all the vowels from the given string.

EXAMPLE:

Input ="abcdefghijklmnopqrstuvwxyz"

Output="bcdfghjklmnpqrstvwxyz"

```
char * removeVowel(char *str)
```

```
{
```

```
    // write your code here
```

```
}
```

Ans: check character by character. If current character is a vowel shift all the following characters one position left.

```
#include <stdio.h>

char * removeVowel(char *str)

{
    int i,j,k;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u' ||
           str[i]=='A' || str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U')
        {
            for(k=i;str[k]!='\0';k++)
                str[k]=str[k+1];
        }
    }
    return str;
}

int main(void) {
    char str[]="abcdef", *s;
    s=removeVowel(str);
    printf("%s",s);
    return 0;
}
```

## Q) Pyramid of alphabets

a  
bcd  
efghi  
jklmnop

```
void printPattern(int n){  
    // write your code here  
}
```

Ans: for every line do two things.

- 1) print spaces
- 2) print alphabets

```
#include<stdio.h>  
  
void printPattern(int n){  
    int i,j;  
    char ch='a';  
    for(i=1;i<=n;i++) {  
        for(j=1;j<=n-i;j++)  
            printf(" ");  
        for(j=1;j<=2*i-1;j++)  
            printf("%c",ch++);  
        printf("\n");  
    }  
}
```

```
int main(void)
{
    printPattern(3);
    return 0;
}
```

Q) You have to encrypt a non-empty string phrase.

The encryption adds a 'cyclic shift' to each letter where the value of this 'cyclic shift' is decided by the position of the letter from the end of its word. The shift value for each letter of a word is its index value (starting from 0) from the right-most character of the word.

EXAMPLE:

The shift values in 'yum feed' will be

yum: m->0, u->1, y->2

feed: d->0, e->1, e->2, f->3

which gives the encryption avm igfd

Here, adding the shift with value 0 to letter 'm' gives 'm' + 0 = m;

values 1 to 'u' gives 'u' + 1 = v and values 2 to 'y' gives 'y' + 2 = a

Note that the shift wraps around on reaching the end of the alphabets, i.e., the shift values for 'y' as shown above is 'a'.

```
char* encryption(char* str)
```

```
{
```

```
    // write your CODE here
```

```
}
```

Ans: separate each word and perform substitution with the respective character in the string

```
#include<stdio.h>
#include<string.h>
char* encryption(char* str) {
    int i,j=0,k,ws=0,val;
    char word[20];
    int len=strlen(str);
    for(i=0;i<=len;i++){
        if(str[i]==' ' || str[i]=='\0')
        {
            val=0;
            for(k=j-1;k>=0;k--)
            {
                word[k]=word[k]+val;
                val++;
                if(word[k]>122)
                    word[k]='a'+(word[k]-123);
                str[ws+k]=word[k];
            }
            ws=i+1;
            j=0;
        }
        else
            word[j++]=str[i];
    }
    return str;
}
```

```
int main()
{
    char str[]="yum feed",*s;
    s=encryption(str);
    printf("%s",s);
    return 0;
}
```

Q) The LeastRecentlyUsed(LRU) cache algorithm exits the element from the cache(when it's full) that was least recently used. After an element is requested from the cache, it should be added to the cache(if not already there) and considered the most recently used element in the cache.

Initially, the cache is empty. The input to the function LruCountMiss shall consist of an integer max\_cache\_size, an array pages and its length len. The function should return an integer for the number of cache misses using the LRU cache algorithm.

Assume that the array pages always has pages numbered from 1 to 50.

TEST CASE1:

Input: 3 16 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0

Expected Return Value: 11

TESTCASE 2:

Input: 2 9 2 3 1 3 2 1 4 3 2

Expected Return Value: 7

```
int LruCountMiss(int max_cache_size,int *pages,int pages_len)
{
    // write your code here
}
```

```
#include <stdio.h>

int LruCountMiss(int max_cache_size,int *pages,int pages_len)

{

int i,j,cache[100],k,pageincache=0,misscount=0;

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

cache[i]=-1;

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

{

pageincache=0;

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

{

if(pages[i]==cache[j])

{

pageincache=1;

for(k=j;k<max_cache_size;k++)

{

cache[k]=cache[k+1];

}

cache[max_cache_size-1]=pages[i];

break;

}

}

misscount++;

}

}

}
```

```
if(pageincache==0)
{
    misscount++;
    for(k=0;k<max_cache_size;k++)
    {
        cache[k]=cache[k+1];
    }
    cache[max_cache_size-1]=pages[i];
}
return misscount;
}
```

```
int main()
{
    int misscount, pages[]={7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0};
    misscount=LruCountMiss(3, pages, 16);
    printf("%d ", misscount);
}
```

Q) The methods `GetDigitSum(int *arr,int len)` accepts an integers array arr. it is supposed to calculate the sum of digits of the smallest elements in the input array. It returns 1 if the calculated sum is even and returns 0 otherwise. However there is a compilation error in the code your task is to fix it so that the program works for all the input values.

Note: The methods `getdigitSum` uses another method `getSum(int num)` which returns the sum of the digits of the input number `num`. Write code for that function

```
#include<stdio.h>

int getDigitSum(int *arr,int len)

{
    int result;
    for(int i=0;min=arr[0];i<len;i++)
    {
        if(arr[i]<min)
            min=arr[i];
    }
    result=getSum(min);
    if(result%2==0)
        return 1;
    else
        min==arr[j];
}
```

```
int getSum(int num){  
    // write your code here  
}  
  
int main(){  
    int a[]={1423,567,897,426,765},s;  
    s=getDigitSum(a,5);  
    printf("%d",s);  
}
```

Ans:

```
#include<stdio.h>  
  
int getSum(int num);  
  
int getDigitSum(int *arr,int len){  
    int result,min,i;  
    for(i=0,min=arr[0];i<len;i++) {  
        if(arr[i]<min)  
            min=arr[i];  
    }  
    result=getSum(min);  
    if(result%2==0)  
        return 1;  
    else  
        return 0;  
}
```

```
int getSum(int num)
{
    int rem,sum=0;
    while(num)
    {
        rem=num%10;
        sum+=rem;
        num/=10;
    }
    return sum;
}
```

```
int main()
{
    int a[]={1423,567,897,426,765},s;
    s=getDigitSum(a,5);
    printf("%d",s);
}
```

Telegram: <https://t.me/campusdrive>

## Cognizant Automata Fix All 31 Solved Questions on 16<sup>th</sup> October 2020

Logical error

Problem (1):

The function/method countDigits return an integer representing the remainder when the given number is divided by the number of digits in it.

The function/method countDigits accepts an argument-num,an integer representing the given number.

The function/method countDigits compiles code successfully but fails to get the desirable result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

---

Test case 1:

Input: 782

Expected return value: 2

Test case 2:

Input: 21340

Expected return value: 0

Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

---

```
int countDigits(int num)
{
    int count=0;
    while(num!=0)
    {
        num=num/10;
        count++;
    }
    return (num%count);
}
```

---

solution:

```
int countDigits(int num)
{
    int count=0,temp=num;
    while(temp!=0)
    {
        temp=temp/10;
        count++;
    }
    return (num%count);
}
```

---

Problem ( 2 ):

Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

The function/ method arrayReverse modify the input list by reversing its element. The function/method arrayReverse accepts two arguments - len an integer representing the length of the list and arr, list of integers representing the input list respectively

For example, if the input list arr is {20 30 10 40 50}.

The function/method arrayReverse compiles code successfully but fails to get the desirable result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

---

Testcase 1:

Input:

4,[4,2,8,6]

Expected return value:

6 8 2 4

Testcase 2:

Input:

3,[11,20,17]

Expected return value:

17 20 11

---

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
void arrayReverse(int len,int *arr)
{
    int i,temp,originalLen=len;
    for(i=0;i<=originalLen/2;i++)
    {
        temp=arr[len-1];
        arr[len-1]=arr[i];
        arr[i]=temp;
        len-=1;
    }
}
```

---

Problem (3 ):

The function/method printCharacterPattern accepts an integer num. It is supposed to print the first num( $0 \leq \text{num} \leq 26$ ) lines of the pattern as shown below:

For example,if num=4, the pattern is

a  
ab  
abc  
abcd

The function/method printCharacterPattern compiles successfully but fails to get desirable output. Fix the logical errors.

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

---

```
void printCharacterPattern(int num)
{
    int i,j;
    char ch='a';
    char print;
    for(i=0;i<num;i++)
    {
        print=ch;
        for(j=0;j<=i;j++)
        {
            printf("%c",ch++);
        }
        printf("\n");
    }
}
```

---

Solution:

```
void printCharacterPattern(int num)
{
    int i,j;
    char ch='a';
    char print;
    for(i=0;i<num;i++)
    {
        print=ch;
        for(j=0;j<=i;j++)

```

**Telegram: <https://t.me/campusdrive>**

Telegram: <https://t.me/campusdrive>

```
{  
    printf("%c",print++);  
}  
printf("\n");  
}  
}
```

---

Probelm ( 4 ):

The function/method removeElement prints space seperated integer that remains after removing the integer at the given index from the input list.

The function/method removeElement accepts three arguments, size an integer representing the size of input list, indexValue an integer representing given index and inputList, a list of integers representing the input list.

The function/method removeElement compiles code successfully but fails to get the desirable result for some test cases due to incorrect implementation of function/method removeElement. Your task is to fix the code so that it passes all the test cases.

Note: zero based indexing is followed to access list elements

---

Test case 1:

Status: wrong

Expected:

1 2 3 5 6 7 8 9

Returned

1 2 3 4 4 6 6 8

Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

Test case 1:

Status: correct

Expected:

11 23 12 34 54 32

Returned

11 23 12 34 54 32

---

```
void removeElement(int size,int indexValue,int *inputList)
```

```
{
```

```
    int i,j;
```

```
    if(indexValue<size)
```

```
{
```

```
        for(i=indexValue;i<size;i++)
```

```
{
```

```
            inputList[i]=inputList[i++];
```

```
}
```

```
        for(i=0;i<size-1;i++)
```

```
{
```

```
            printf("%d",inputList[i]);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    for(i=0;i<size;i++)
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
{  
    printf("%d",inputList[i]);  
}
```

---

solution:

```
void removeElement(int size,int indexValue,int *inputList)  
{  
    int i,j;  
    if(indexValue<size)  
    {  
  
        for(i=indexValue;i<size;i++)  
        {  
            inputList[i]=inputList[i+1];//1 2 3 4 5 6 7 8 9  
        }  
  
        for(i=0;i<size-1;i++)  
        {  
            printf("%d",inputList[i]);  
        }  
    }  
    else  
    {  
        for(i=0;i<size;i++)  
        {  
            printf("%d",inputList[i]);  
        }  
    }  
}
```

**Telegram: <https://t.me/campusdrive>**

}

---

Probelm ( 5 ):

The function calculateMatrixSum(int \*\* matrix, int m, int n) accepts a two dimensional array matrix of dimensions m, n as input and returns the sum of odd elements whose ith and jth index are same.

The function compiles successfully but fails to return the desired result for some test cases

PROGRAM

```
int calculateMatrixSum(int rows,int columns,int ** matrix)
{
    int i,j,sum=0;
    if((row>0)&&(column>0))
    {
        for(i=0;i<row;i++)
        {
            sum=0;
            for(j=0;j<column;j++)
            {
                if(i==j)
                {
                    if(matrix[i][j]/2!=0)
                        sum+=matrix[i][j];
                }
            }
        }
    }
    return sum;
}
```

**Telegram: <https://t.me/campusdrive>**

```
    }  
    else  
        return sum;  
}  
-----  
solution:
```

```
int calculateMatrixSum(int rows,int columns,int ** matrix)  
{  
    int i,j,sum=0;  
    if((row>0)&&(column>0))  
    {  
        for(i=0;i<row;i++)  
        {  
  
            for(j=0;j<column;j++)  
            {  
                if(i==j)  
                {  
                    if(matrix[i][j]%2!=0)  
                        sum+=matrix[i][j];  
                }  
            }  
        }  
        return sum;  
    }  
    else  
        return sum;  
}
```

**Telegram: <https://t.me/campusdrive>**

Problem (6):

The function/method sortArray modify the input list by sorting its elements in descending order. It accepts two arguments-len,representing the length of the list and arr, a list of integers representing the input list respectively.

It compiles successfully but fails for some test case. Fix the code..

---

```
void sortArray(int len,int *arr)
{
    int i,max,location,j,temp;
    for(i=0;i<len;i++)
    {
        max=arr[i];
        location=i;
        for(j=i;j<len;j++)
        {
            if(max>arr[j])
            {
                max=arr[j];
                location=j;
            }
        }
        temp=arr[i];
        arr[i]=arr[location];
        arr[location]=temp;
    }
}
```

**Telegram: <https://t.me/campusdrive>**

---

solution:

```
void sortArray(int len,int *arr)
{
    int i,max,location,j,temp;
    for(i=0;i<len;i++)
    {
        max=arr[i];
        location=i;
        for(j=i;j<len;j++)
        {
            if(max<arr[j])
            {
                max=arr[j];
                location=j;
            }
        }
        temp=arr[i];
        arr[i]=arr[location];
        arr[location]=temp;
    }
    for(i=0;i<len;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

---

Problem ( 7 ):

**Telegram: <https://t.me/campusdrive>**

The function/method replaceValues is modifying the input list in such a way - if the length of the input list is odd, then all the elements of the input list are supposed to be replaced by 1s and in case it is even, the elements should be replaced by 0s.

For example, given the input list[0,1,2], the function will modify the input list like [1 1 1]. It accepts two arguments size, an integer representing the size of input list and inputList, a list of integers.

The function complies successfully but fails to return the desired results due to logical errors

Your task is to debug the program to pass all the test cases

---

```
void replaceValues(int size,int *inputList)
{
    int i,j;
    if(size%2==0)
    {
        i=0;
        while(i<size)
        {
            inputList[i]=0;
            i+=2;
        }
    }
    else
    {
        j=0;
        while(j<size)
        {
```

**Telegram: <https://t.me/campusdrive>**

```
    inputList[j]=1;
    j+=2;
}
}

-----

```

solution:

```
void replaceValues(int size,int *inputList)
```

```
{
    int i,j;
    if(size%2==0)
    {
        i=0;
        while(i<size)
        {
            inputList[i]=0;
            i+=1;
        }
    }
    else
    {
        j=0;
        while(j<size)
        {
            inputList[j]=1;
            j+=1;
        }
    }
}
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
    }  
}  
-----
```

Problem ( 8 ):

The function productMatrix accepts a two dimensional array matrix of dimensions m, n as input and returns the sum of odd elements whose ith and jth index are same.

The function compiles line but fails to return the desired result for some test cases

PROGRAM

```
int productMatrix(int rows,int columns,int **matrix)  
{  
    int result=0;  
    for(i=0;i<row;i++)  
    {  
        for(j=0;j<column;j++)  
        {  
            if((i==j) | | matrix[i][j]%2!=0)  
                result*=matrix[i][j];  
        }  
    }  
    if(result<=1)      return 0;  
  
    else  
        return result;  
}
```

**Telegram: <https://t.me/campusdrive>**

solution:

```
int productMatrix(int rows,int columns,int **matrix)
{
    int result=0,flag=0;
    for(i=0;i<row;i++)
    {
        for(j=0;j<column;j++)
        {
            if((i==j) && matrix[i][j]%2!=0)
            {
                result*=matrix[i][j];
                flag=1;
            }
        }
    }
    if(flag==0)
        return 0;
    else
        return result;
}
```

---

Problem ( 9 ):

The function maxReplace is supposed to replace every element of the input array arr with the maximum element of arr.

**Telegram: <https://t.me/campusdrive>**

The function complies unsuccessfully due to compilation errors.

Your task is to debug the program to pass all the test cases

PROGRAM:

```
void maxReplace(int size,int &inputList)
{
    int i;
    if(size>0)
    {
        int max=inputList[0];
        for(i=0;i<size;i++)
        {
            if(max<inputList[i])
            {
                max=inputList[i];
            }
        }
        for(i=0;i<size;i++)
        {
            inputList[i]=max
            printf("%d",inputList[i]);
        }
    }
}
```

---

solution:

```
void maxReplace(int size,int *inputList)
```

**Telegram: <https://t.me/campusdrive>**

Telegram: <https://t.me/campusdrive>

```
{  
    int i,max;  
    if(size>0)  
    {  
        max=inputList[0];  
        for(i=0;i<size;i++)  
        {  
            if(max<inputList[i])  
            {  
                max=inputList[i];  
            }  
        }  
    }  
    for(i=0;i<size;i++)  
    {  
        inputList[i]=max;  
        printf("%d",inputList[i]);  
    }  
}
```

---

Problem ( 10 ):

The function/method sumElement return an integer representing the sum of the elements in the input array that are greater than twice the input number K and present at the even index. It accepts three arguments-size of array, numK representing input number and inputarray list of integers.

It compiles successfully but fails to return the desirable result.

---

```
int sumElement(int size, int numK,int *inputArray)  
{
```

Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

```
int i,sum=0;
for(i=0;i<size;i++)
{
    if(inputArray[i]>2*numK && i/2==0)
    {
        sum=inputArray[i];
    }
}
return sum;
}
```

---

solution:

```
int sumElement(int size, int numK,int *inputArray)
{
    int i,sum=0;
    for(i=0;i<size;i++)
    {
        if(inputArray[i]>2*numK && i%2==0)
        {
            sum+=inputArray[i];
        }
    }
    return sum;
}
```

---

Problem (11):

Telegram: <https://t.me/campusdrive>

## Telegram: <https://t.me/campusdrive>

The function/method manchester: for each element in the input array arr, a counter is incremented if the bit arr[i] is same as arr[i-1]. Then the increment counter value is added to the output array to store the result.

If the bit arr[i] and arr[i-1] are different, then 0 is added to output array. for the first bit in the input array, assume its previous bit to be 0. For example if arr is {0,1,0,0,1,1,1,0}, then it should print 1 0 0 2 0 3 4 0.

It accepts two arguments-size and arr-list of integers. Each element represents a bit 0 or 1.

It compiles successfully but fails to print the desirable result. Fix the code.

```
void manchester(int size,int *arr)
{
    bool result;
    int *res=(int *)malloc(sizeof(int)*size);
    int count=0;
    for(int i=0;i<size;i++)
    {
        if(i==0)
            result=(arr[i]==0);
        else
            result=(arr[i]==arr[i-1]);
        res[i]=(result)?0:(++count);
    }
    for(int i=0;i<size;i++)
    {
        printf("%d",res[i]);
    }
}
```

## Telegram: <https://t.me/campusdrive>

solution:

```
void manchester(int size,int *arr)
{
    bool result;
    int res[size];
    int count=0;
    for(int i=0;i<size;i++)
    {
        if(i==0)
            result=(arr[i]==1);
        else
            result=!(arr[i]==arr[i-1]);
        res[i]=(result)?0:(++count);
    }
    for(int i=0;i<size;i++)
    {
        printf("%d ",res[i]);
    }
}
```

---

Problem (12):

The function/method printFibonacci accepts an integer num, representing a number. The function/method printFibonacci prints first num numbers of fibonacci series.

For example,given input 5, the function should print the string "01123"(without quotes).

## Telegram: <https://t.me/campusdrive>

It compiles successfully but fails to give the desirable result for some test cases. Debug the code.

---

```
void printFibonacci(int num)
{
    long num1=0;
    long num2=1;
    for(int i=1;i<num;++i)
    {
        printf("%ld",num1);
        long sum=num1+num2;
        num2=sum;
        num1=num2;
    }
}
```

---

Solution:

```
void printFibonacci(int num)
{
    long num1=0;
    long num2=1;
    for(int i=1;i<=num;++i)
    {
        printf("%ld ",num1);
        long sum=num1+num2;
        num1=num2;
        num2=sum;
    }
}
```

## Telegram: <https://t.me/campusdrive>

```
    }  
}  
-----
```

Problem (13):

The function/method selectionSortArray performs an in-place selection sort on the given input list which will be sorted in ascending order.

It accepts two arguments-len,an integer representing the length of input list and arr, a list of integers representing the input list respectively.

It compiles successfully but fails to get the desired result .

```
void selectionSortArray(int len,int *arr)  
{  
    int x=0,y=0;  
    for(x=0;x<len;x++)  
    {  
        int index_of_min=x;  
        for(y=x;y<len;y++)  
        {  
            if(arr[index_of_min]>arr[y])  
            {  
                index_of_min=y;  
            }  
        }  
        int temp=arr[x];  
        arr[x]=arr[index_of_min];  
        arr[index_of_min]=temp;  
    }  
}
```

Telegram: <https://t.me/campusdrive>

}

---

solution:

```
void selectionSortArray(int len,int *arr)
{
    int x=0,y=0;
    for(x=0;x<len;x++)
    {
        int index_of_min=x;
        for(y=x;y<len;y++)
        {
            if(arr[index_of_min]>arr[y])
            {
                index_of_min=y;
            }
        }
        int temp=arr[x];
        arr[x]=arr[index_of_min];
        arr[index_of_min]=temp;
    }
    for(x=0;x<len;x++)
    {
        printf("%d ",arr[x]);
    }
}
```

---

Probelm (14):

Telegram: <https://t.me/campusdrive>

## Telegram: <https://t.me/campusdrive>

The function/method descendingSortArray performs an in-place sort on the given input list which will be sorted in descending order.

It accepts two argument-len of an array and array of elements.

It compiles successfully but fails to get the desirable output.

---

```
void descendingSortArray(int len, int *arr)
{
    int small, pos, i, j, temp;
    for(i=0; i<=len-1; i++)
    {
        for(j=i; j<len; j++)
        {
            temp=0;
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```

---

solution:

## Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

```
void descendingSortArray(int len, int *arr)
{
    int small, i,j,temp;
    for(i=0;i<=len-1;i++)
    {
        for(j=i+1;j<len;j++)
        {
            temp=0;
            if(arr[i]<arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for(i=0;i<=len-1;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

---

problem (15):

The function/method patternPrint accepts an argument num,an integer.

The function/method patternPrint prints num lines in the following pattern.

For example, num=4, the pattern should be

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
1
11
111
1111
```

It compiles successfully but fails to print desirable result.

---

```
void patternPrint(int num)
{
    int print=1,i,j;
    for(i=0;i<num;i++)
    {
        for(j=0;j<=i;j++);
        {
            printf("%d", print);
        }
        printf("\n");
    }
}
```

---

solution:

```
void patternPrint(int num)
{
    int print=1,i,j;
    for(i=0;i<num;i++)
    {

```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
for(j=0;j<=i;j++)  
{  
    printf("%d", print);  
}  
printf("\n");  
}  
-----
```

Problem (16):

Lisa always forgets her birthday which is on the 5th of july. So,develop a function/method which will be helpful to remember her birthday.

It return an integer "1" if it is her birthday else return 0. It accepts two argument-month of her birthday, day of her birthday.

It compiles successfully but fails to return the desirable output.

```
-----  
int checkBirthday(char *month,int day)  
{  
    if(strcmp(month,"July"))|| (day==5))  
        return 1;  
    else  
        return 0;  
}
```

-----  
solution:

**Telegram: <https://t.me/campusdrive>**

```
int checkBirthday(char *month,int day)
{
    if(strcmp(month,"July"))&&(day==5))
        return 1;
    else
        return 0;
}
```

---

Problem (17):

The function/method sameElementCount returns an integer representing the number of elements of the input list which are even numbers and equal to the element to its right. For eg, if the input list is [4 4 1 8 4 1 1 2 2] then the function should return the output 3 as it has three similar groups i.e., (4,4), (4,4), (2,2)

It compiles successfully but fails to return desirable output.

---

```
int sameElementCount(int size,int inputList)
{
    int i,count=0;
    for(i=0;i<size;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
        {
            count++;
        }
    }
    return count;
}
```

Telegram: <https://t.me/campusdrive>

}

---

solution:

```
int sameElementCount(int size,int *arr)
{
    int i,count=0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0) && inputList[i]==inputList[i+1])
        {
            count++;
        }
    }
    return count;
}
```

---

Problem (18):

The function/method manchester: for each element in the input array arr, a counter is incremented if the bit arr[i] is same as arr[i-1]. Then the element of the list is 0, otherwise 1.

for the first bit in the input array, assume its previous bit to be 0. For example if arr is {0,1,0,0,1,1,1,0}, then it should print {0,1,1,0,1,0,0,1}

It accepts two arguments-size and arr-list of integers. Each element represents a bit 0 or 1.

It compiles successfully but fails to print the desirable result. Fix the code.

---

Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

```
void manchester(int len,int *arr)
{
    int res[len];
    res[0]=arr[0];
    for(int i=1;i<len;i++)
    {
        res[i]=(arr[i]==arr[i-1]);
    }
    for(int i=0;i<len;i++)
    {
        printf("%d",res[i]);
    }
}
```

---

solution:

```
void manchester(int len,int *arr)
{
    int res[len];
    res[0]=arr[0];
    for(int i=1;i<len;i++)
    {
        res[i]=!(arr[i]==arr[i-1]);
    }
    for(int i=0;i<len;i++)
    {
        printf("%d",res[i]);
    }
}
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
    }  
}  
-----
```

problem (19):

The function/method patternPrint accepts an argument num,an integer.

The function/method patternPrint prints num lines in the following pattern.

For example, num=4, the pattern should be

```
11  
1111  
111111  
11111111
```

It compiles successfully but fails to print desirable result.

---

```
void drawPrintPattern(int num)  
{  
    int i,j,print=1;  
    for(i=1;i<=num;i++)  
    {  
        for(j=1;j<=2*i;j++)  
        {  
            printf("%d",print);  
        }  
    }
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
    printf("\n");
}
-----solution:
```

```
void drawPrintPattern(int num)
{
    int i,j,print=1;
    for(i=1;i<=num;i++)
    {
        for(j=1;j<=2*i;j++)
        {
            printf("%d",print);
        }
        printf("\n");
    }
}
-----
```

Problem (20):

The function/method countOccurrence return integer representing the count of occurrence of given value in input list.

It accepts three arguments-len,value,arr list.

It compiles successfully but fails to return the desirable result.

```
-----  
int countOccurrence(int len, int value,int *arr)  
{
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
int i=0,count=0;
while(i<len)
{
    if(arr[i]==value)
    {
        count+=1;
    }
}
return count;
```

-----solution:

```
int countOccurence(int len, int value,int *arr)
{
    int i=0,count=0;
    while(i<len)
    {
        if(arr[i]==value)
        {
            count+=1;
        }
        i++;
    }
    return count;
}
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

Complete the code:

Problem (21):

You are given predefined structure Time containing hour, minute and second as members. A collection of functions/methods for performing some common operations on times is also available. You must use these functions/methods to calculate and return the difference.

It accepts two arguments time1 and time 2 representing two times and is supposed to return an integer represents the difference in the number of seconds.

You must complete the code so that it passes all the test cases.

Note:

The following structure is used to represent the time and is already implemented in the default code.

```
typedef struct
{
    int hour,
    int minute,
    int second;
}Time;

int Time_compareTo(const Time* tim1,const Time* time2)
{
    /* Return 1, if time1>time2
    Return -1 if time1<time2
    or, return 0, if time1==time2
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

This can be called as

```
*if time1 and time2 are two times then  
*Time_compareTo(time1,time2);*/  
}
```

```
void Time_addSecond(Time * time)  
{  
    /* Add one second in the time;  
  
}
```

---

```
int difference_in_times(Time *time1,Time * time2)  
{  
    //write code  
}
```

solution:

```
#include<stdio.h>
```

```
typedef struct  
{
```

```
    int hour;
```

```
    int minute;
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
int second;  
}Time;  
  
int Time_compareTo(const Time*,const Time*);  
int difference_in_times(Time*,Time*);  
int t1s,t2s;  
int main()  
{  
Time t1,t2;  
t1.hour=11;t1.minute=43;t1.second=50;  
t2.hour=9;t2.minute=27;t2.second=57;  
t1s=t1.hour*60*60+t1.minute*60+t1.second;  
t2s=t2.hour*60*60+t2.minute*60+t2.second;  
printf("difference of times duration:%d",difference_in_times(&t1,&t2));  
}  
  
int Time_compareTo(const Time* time1,const Time* time2)  
{  
if(t1s>t2s) return 1;  
else return -1;  
return 0;  
}  
  
int difference_in_times(Time* time1,Time* time2)  
{  
if(Time_compareTo(time1,time2)==1) return t1s-t2s;  
else return t2s-t1s;  
return 0;  
}
```

**Telegram: <https://t.me/campusdrive>**

Problem (22):

The function `findMaxElement(int *arr1,int len1,int *arr2,int len2)` accepts two integer arrays `arr1, arr2` of length `len1, len2` respectively. It is supposed to return the largest element in both the input arrays.

Another function `sortArray(int *arr,int len)` sorts the input array `arr` of length `len` in ascending order and returns the sorted array.

Your task is to use `sortArray(int *arr,int len)` function and complete the code in `findMaxElement(int *arr1,int len1,int *arr2,int len2)` so that it passes all testcases"

---

```
int *sortArray(int len,int *arr)
{
    int i=0,j=0,temp=0;
    for(i=0;i<len;i++)
    {
        for(j=i+1;j<len;j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```

**Telegram: <https://t.me/campusdrive>**

```
return arr;  
}  
  
int findMaxElement(int len1,int *arr1,int len2,int *arr2)  
{  
    //write code here  
}
```

---

Solution:

```
int *sortArray(int len,int *arr)  
{  
    int i=0,j=0,temp=0;  
    for(i=0;i<len;i++)  
    {  
        for(j=i+1;j<len;j++)  
        {  
            if(arr[i]>arr[j])  
            {  
                temp=arr[i];  
                arr[i]=arr[j];  
                arr[j]=temp;  
            }  
        }  
    }  
    return arr;  
}
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
int findMaxElement(int len1,int *arr1,int len2,int *arr2)
{
    int sort_array1[len1]=sortArray(int len1,int *arr1);
    int sort_array2[len2]=sortArray(int len2,int *arr2);
    if(sort_array1[len1-1]>sort_array2[len2-1])
    {
        printf("%d",sort_array1[len1-1]);
    }
    else
    {
        printf("%d",sort_array2[len2-1]);
    }
}
```

---

Problem (23):

The function allExponent(int base,int exponent) accepts two integers base and exponent as inputs. It is supposed to calculate and return the exponentiation of base raised to power exponent for all input values.

The function positiveExponent(int base,int exponent) will return the exponentiation value if exponent value is positive. Try to complete the code such that allExponent should return value if exponent value is negative.

PROGRAM

```
float allExponent( int base,int exponent )
{
```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
float res=1;  
if(exponent >=0)  
{  
    Res = (float)positiveExponent(base,exponent)  
}  
Else  
{  
    //write ur code here  
}  
return res;  
}  
-----
```

solution:

```
float allExponent( int base,int exponent )  
{  
float res=1;  
if(exponent >=0)  
{  
    res = (float)positiveExponent(base,exponent);  
}  
else  
{  
    //write ur code here  
    res = 1/((float)positiveExponent(base,-(exponent)));  
}  
return res;  
}  
int positiveExponent(int base,int exponent)
```

**Telegram: <https://t.me/campusdrive>**

Telegram: <https://t.me/campusdrive>

```
{  
    int sol=pow(base,exponent);  
    return sol;  
}
```

---

Problem (24):

You are given a predefined structure Point and related functions.

The function/method isTriangle which accepts three points P1,P2,P3 as inputs and checks whether the given three points form a triangle.

If they form a triangle the function return 1 else 0.

```
typedef struct point  
{  
    int X;  
    int Y;  
}Point;  
  
double Point_calculateDistance(Point *point1,Point *point2)
```

```
{  
    //returns the distance between point 1, point2. It can be called as  
    Point_calculateDistance(P1,P2);  
}
```

---

```
int isTriangle(Point *P1,Point *P2,Point *P3)  
{  
    //write code
```

Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

}

---

solution:

```
int isTriangle(Point *P1,Point *P2,Point *P3)
{
    int a=P1.X*(P2.Y-P3.Y)+P2.X*(P3.Y-P1.Y)+P3.X*(P1.Y-P2.Y);

    if(a==0)
        return 0;
    else
        return 1;
}
```

---

Problem (25):

Charlie has a magic mirror. The mirror shows right rotated versions of a given word.

To generate different right-rotations of a word, write the word in a circle in clockwise order, then start reading from any given character in clockwise order till you have covered all the characters. For example: In the word "sample", if we start with 'p', we get the right rotated word as "plesam". There are six such right rotations of "sample" including itself.

The inputs to the function isSameReflection consists of two strings, word1 and word2.

The function returns 1 if word1 and word2 are right rotations of the same word and -1 if they are not. Both word 1 and word2 will strictly contain characters between 'a'-'z' (lower case letters).

-->Useful commands:

Telegram: <https://t.me/campusdrive>

## Telegram: <https://t.me/campusdrive>

strlen() is used to calculate the length of the string.

The statement -int len = strlen(str)

returns the lenght of the srting str

---

```
using namespace std;
```

```
int isSameReflection(char *word1,char *word2)
```

```
{
```

```
    //write code here
```

```
    int size1=strlen(word1);
```

```
    int size2=strlen(word2);
```

```
    char *temp;
```

```
    void *ptr;
```

```
    if(size1!=size2)
```

```
        return -1;
```

```
    temp[0]="";
```

```
    strcat(temp,word1);//concat empty string with word1(sample)
```

```
    strcat(temp,word1);//concat sample along with sample again(samplesample)
```

```
    //temp=samplesample word2=zlesam
```

ptr=strstr(temp,word2);//word2 is there in temp and return index of first char of word2 in temp, if word2 is not a substring it will return null.

```
    if(ptr!=null)
```

```
        return 1;
```

```
    else
```

```
        return -1;
```

## Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

}

----c++----

```
int isSameReflection(String str1, String str2)
{
    if(str1.length() != str2.length())
        return false;
    string temp = str1 + str1;
    if(temp.find(str2) != string::npos)
    {
        return 1;
    }
    else return -1;
}
```

---

Problem (26):

You are given a predefined structure Point and related functions.

The function/method isRightTriangle which accepts three points P1, P2, P3 as inputs and checks whether the given three points form a right angled triangle.

If they form a right triangle the function return 1 else 0.

```
typedef struct point
{
    int X;
```

Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

```
int Y;  
}Point;  
  
double Point_calculateDistance(Point *point1,Point *point2)  
{  
    //returns the distance between point 1, point2. It can be called as  
    Point_calculateDistance(P1,P2);  
}
```

---

solution:

```
int isRightTriangle(Point *P1,Point *P2,Point *P3)  
{  
    double d1,d2,d3;  
    d1=Point_calculateDistance(P1,P2);  
    d2=Point_calculateDistance(P2,P3);  
    d3=Point_calculateDistance(P3,P1);  
  
    if(pow(d1,2)+pow(d2,2)==pow(d3,2) ||  
       pow(d2,2)+pow(d3,2)==pow(d1,2) ||  
       pow(d3,2)+pow(d1,2)==pow(d2,2))  
    {  
        return 1;  
    }  
    else  
        return 0;  
}
```

---

Problem (27):

Telegram: <https://t.me/campusdrive>

## Telegram: <https://t.me/campusdrive>

The function/method median accepts two arguments-size and inputList,an integer representing the length of a list and a list of integers respectively.

It is supposed to calculate and return an integer representing the median of elements in the input list. However the function/method median works only for odd-length lists because of incomplete code.

you must complete the code for even-length as well.

```
int quick_select(int*inputList,int start_index,int end_index,int median_order)
{
    //It calculate the median value
}
```

---

solution:

```
float median(int size,int *inputList)
{
    int start_index=0;
    int end_index=size-1;
    float res=-1;
    if(size%2!=0)
    {
        int med_1=size/2;
        int med_2=(size/2)+1;

        res=((float)quick_select(inputList,start_index,end_index,med_1));
    }
    else
```

## Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

```
{  
    //enter your code here  
  
    res=((float)quick_select(inputList,start_index,end_index,med_1)+(float)quick_select(inputList,st  
art_index,end_index,med_2))/2;  
}  
  
return res;  
}
```

---

Compilation error:

Problem (28):

The function/method matrixSum returns an integer representing the sum of elements of the input matrix.

The function/method matrixSum accepts three arguments- rows, an integer representing the number of rows ,columns an integer representing number of columns, matrix is going to be the input matrix.

The function/method matrixSum has some compilation error. Rectify the error and get the desired output:

---

```
int matrixSum(int rows,int columns, int **matrix)  
{  
    int i,j,sum=0;  
    for(i=0;i<rows;i++)  
    {
```

Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

```
for(j=0;j<columns;j++)  
{  
    sum+=matrix(i)(j);  
}  
}  
return sum;  
}
```

---

solution:

```
int matrixSum(int rows,int columns, int **matrix)  
{  
    int i,j,sum=0;  
    for(i=0;i<rows;i++)  
    {  
        for(j=0;j<columns;j++)  
        {  
            sum+=matrix[i][j];  
        }  
    }  
    return sum;  
}
```

---

Problem (29):

The function/method countElement return an integer representing the number of elements in the input list which are greater than twice the input number K.

**Telegram: <https://t.me/campusdrive>**

## Telegram: <https://t.me/campusdrive>

The function/method accepts three arguments-size, an integer representing the size of input list, numK an integer representing the input number K and inputList, a list of integers representing the input list, respectively.

The function/method countElement compiles unsuccessfully due to systematical error. Fix the code so that it passes all test cases.

---

```
int countElement(int size,int num,int *inputList)
{
    int i,co-unt=0;
    for(i=0;i<size;i++)
    {
        if(inputList[i]>2numK)
        {
            co-unt+=1;
        }
    }
    return co-unt;
}
```

---

solution:

```
int countElement(int size,int num,int *inputList)
{
    int i,count=0;
    for(i=0;i<size;i++)
    {
        if(inputList[i]>2*numK)
```

## Telegram: <https://t.me/campusdrive>

Telegram: <https://t.me/campusdrive>

```
{  
    count+=1;  
}  
}  
return count;  
}
```

---

Problem (30):

The function/method deleteNonRepeat accpets two arguments-size, an integer representing the size of the list and inputList,representing the list of integers.

It removes the non-repeated integers from the inputList and prints the remaining repeated integers seperated by space from inputList.

If all the elements are unique then the function should not print anything.

For example, for the given inputList{2,3,2,2,5,6,6,7}, th expected output is 2 2 2 6 6 .

It compiles unsuccessfully due to syntactical error. Debug the code.

---

```
void deleteNonRepeat(int size,int *inputList)  
{  
    int count=0,i,j;  
    int counter[size];  
    for(i=0;i<size;i++)  
    {  
        counter[i]=1;  
    }
```

Telegram: <https://t.me/campusdrive>

**Telegram: <https://t.me/campusdrive>**

```
}

for(i=0;i<size;i++)
{
    for(j=i+1;j<size;j++)
    {
        if(inputList[i]==inputList[j])
        {
            counter[i]++;
            counter[j]++;
        }
    }
}

for(i=0;i<size;i++)
{
    if(counter[i]>1)
        count++;

j=0;
for(i=0;i<size;i++)
{
    if(counter[i]>1)
        printf("%d",inputList[i]);
}
-----
```

solution:

```
void deleteNonRepeat(int size,int *inputList)
```

```
{
    int count=0,i,j;
    int counter[size];
    for(i=0;i<size;i++)

```

**Telegram: <https://t.me/campusdrive>**

**Telegram: <https://t.me/campusdrive>**

```
{  
    counter[i]=1;  
}  
for(i=0;i<size;i++)  
{  
    for(j=i+1;j<size;j++)  
    {  
        if(inputList[i]==inputList[j])  
        {  
            counter[i]++;  
            counter[j]++;  
        }  
    }  
}  
for(i=0;i<size;i++)  
if(counter[i]>1)  
    count++;  
j=0;  
for(i=0;i<size;i++)  
if(counter[i]>1)  
    printf("%d",inputList[i]);  
}
```

---

Problem (31):

The function/method multiplyNumber returns an integer representing the multiplicative product of the maximum two of three numbers. It accepts three integers- numA, numB, and NumC, representing the input numbers.

**Telegram: <https://t.me/campusdrive>**

Telegram: <https://t.me/campusdrive>

It compiles unsuccessfully due to syntactical error. Debug the code.

---

```
int multiplyNumber(int numA,int numB,int numC)
{
    int result,min,max,mid;
    max=(numA>numB)?numA>numC?numA:numc):((numB>numC)?numB:numC);
    min=(numA<numB)?((numA<numC)?numA:numc):((numB<numC)?numB:numC);
    mid=(numA+numB+numC)-(min+max);
    result=(max*int mid);
    return result;
}
```

---

solution:

```
int multiplyNumber(int numA,int numB,int numC)
{
    int result,min,max,mid;
    max=(numA>numB)?((numA>numC)?numA:numc):((numB>numC)?numB:numC);
    min=(numA<numB)?((numA<numC)?numA:numc):((numB<numC)?numB:numC);
    mid=(numA+numB+numC)-(min+max);
    result=(max* mid);
    return result;
}
```

---

Telegram: <https://t.me/campusdrive>

"He who has never learned to obey cannot be a good leader" Write a response that describes the traits of a good leader. To what extent do you agree with your statement? Explore the arguments for and against the statement. Max words 400

First of all, confidence is the best quality. A leader must have strong self-confidence. A person lacking in confidence can never be a good leader. A person must be confident enough to ensure others follow him. The leader must have confidence in his decisions and actions. If he is unsure, then how can people have the desire to follow him?

A good leader must certainly inspire others. A leader must be a role model for his followers. Furthermore, he must motivate them whenever possible. Also, in difficult situations, a leader must not lose hope. How can a leader inspire people if he himself is hopeless?

Honesty is another notable quality of a leader. Honesty and Integrity are important to earn the love of followers. Above all, honesty is essential to win the trust of the people. Probably, every Leadership which loses trust is bound to fail. People will not work with full effort due to an immoral leader.

Good communication is a must for a good leader. This is because poor communication means the wrong message to followers. Furthermore, good communication will increase the rate of work. Also, the chances of mistakes by followers will be reduced.

Another important quality is decision making. Above all, if a leader makes poor decisions then other qualities will not matter. Furthermore, good decision making ensures the success of the entire group. If the leader makes poor decisions, then the efforts of followers won't matter.

A good leader must be an excellent innovator. He must display a creative attitude in his work. Most noteworthy, innovation is a guarantee of survival of a group or innovation. Without creative thinking, progress is not possible.

Essay on the gap between the rich and poor.

The gap between the rich and the poor has moved to the extreme ends over the last decade. The rich have continued to gain more wealth and grow richer whilst the poor have remained poor.

The gap between rich and poor depicts the inequality in income distribution between the rich and the poor.

There are various reasons for the increasing gap between the rich and the poor in society. In most cases, these reasons are related. Research study on the gap between the rich and the poor affirms that; culture, innate capability, globalization, education, labor markets, reforms on taxes, government policies, change in technology, gender, racism and differences in wages and incomes as the main causes.

The difference in wages and salaries is the core reason for the growing gap between the rich and the poor. The job salaries are indomitable by the supply and demand in the commercial market. For instance, when the supply of labor is high and demand for working force is low, the wages for the few available job opportunities will be as well low

Besides the market related factors that affect the differences in wages, initiatives facilitated by the government such as tax reforms and policies can also result in an increase or decrease in inequality. Social scientists and policy developers argue on the efficiency and effectiveness of the inequality regulations strategies. Some of the governmental distinctive initiatives which can minimize the difference between the rich and the poor involve; educating the public that enhances the skills of the workforce supply in order to minimize the wage difference related to education factors.

The implementation of tax reforms programs which will impose relatively higher taxes on the rich compared to the unfortunate will be very significant in regulating income differences within the society. Establishment of minimum income legislative initiative will assist in enhancing wages for the unfortunate laborers. An initiative to subsidize product prices will enable consumers to purchase goods and services at relatively low prices. These initiatives will prove to be very vital in reducing the gap between the rich and the poor within the society.

In general, the situation in which only the rich continue to be rich can be reduced if not solved through emphasizing more on education, investing in human capital and economic literacy. The problem has been there for ages and the gap seems to widen by the dawn of each new day.

Check whether triangle is valid or not

```
// C++ program to check if three
// sides form a triangle or not

#include<bits/stdc++.h>

using namespace std;

// function to check if three sides
// form a triangle or not

bool checkValidity(int a, int b, int c)

{
    // check condition

    if (a + b <= c || a + c <= b || b + c <= a)
        return false;

    else
        return true;
}

// Driver function

int main()

{
    int a = 7, b = 10, c = 5;

    if (checkValidity(a, b, c))
```

```
cout << "Valid";  
  
else  
  
cout << "Invalid";  
  
}
```

Has technology become a new addiction essay.

Technology is everywhere and around us. The thing that you put in your pocket while travelling, your mobile phone is the most prime example of technology. GPS navigation, computer, internet, fan, A/C, and every other device we use in our daily lives are also prime examples of technology. Without technology, life would be dull and hard. Nowadays, countries that have more advanced technologies are developing day by day.

It is observed in a survey that people can't live without technology even a single day as they survive on cell phones. People get so addicted to technological devices that they are not able to work without them. Many companies and industries are dependent on these technologies to operate their functions and manufacture goods.

People are so addicted that they prefer to use their mobile phones to communicate rather than communicating face to face. This affects their mental health and also their relationship with friends and family. People spend lots of money on buying new technology as they have advanced and better features.

The Internet is the prime cause of addiction to mobile phones and computers. It was introduced as a medium to help people for searching for information and data but people started misusing the internet which results in addiction. People started spending hours and hours on mobile phones surfing the internet and watching online videos. Nowadays even children have to wear glasses because of weak eyesight due to continuously using mobile phones and computers.

what is maximum number of edges in a graph with n vertices

In a directed graph having  $N$  vertices, each vertex can connect to  $N-1$  other vertices in the graph (Assuming, no self loop). Hence, the total number of edges can be  $N(N-1)$ .

A librarian has to rearrange the library books on a shelf in a proper order at the end of each day. Which of the following sorting techniques should be the librarian's ideal choice?

Insertion Sort

A sorting algorithm traverses through a list, comparing adjacent elements and switching them under certain conditions.

What is this sorting algorithm called?

Bubble sort

A programmer tries to debug a code of 10,000 lines. It is known that there is a logical error in the first 25 lines of the code. What is an efficient way to debug the code?

Use an interpreter for the first 25 lines.

The tight curriculum of our current education system leaves no room for imagination and creativity-essay

I whole heartedly agree that the tight curriculum of our current education system leaves no room for imagination and creativity. The academic pressure is so much on them, they have forgotten their innate creativity and imagination. They are always under the pressure of studies. This pressure is put on them from the beginning of the session. Formative assessment activities of all the subjects, studies and their assignments of all the subjects, unit tests, co-curricular activities, homework, classwork, projects, practical exams, summative examination, date sheet, result, again the new term, etc. So much of pressure on the young growing children! Are we educating them or training them the art of being stressful throughout their childhood?

Even while studying inside class, they feel so burdened; they have come to dislike education. as a result there is hardly any creativity left in our country. Every year all the illustrious prizes for creativity are won by non-Indians or Indians living in advanced countries. Our educationists must rethink and revise the curriculum to encourage creativity and imagination in children --- not rot learning.

With each passing day, education is becoming more difficult and the system is getting stricter. With such heavy rules and regulations in academics, students hardly have the opportunity to dream and imagine. Children are known for their imagination and it is due to imagination that they can explore new creative arenas. This is creating individuals who are extremely methodical but the true meaning of education is somewhat missing. Education teaches us to be more vocal about our likings and disliking and improves our imaginative skill and creativity.

Unfortunately the Indian education system changed so drastically that the talent of students is being measured only based on the marks scored by him and the success of the educational system is being measured by the number of students who scored a centum. If the student couldn't perform well in the exams, he is tagged 'Average student' which is looked at as a disgrace for both the student and the parents. Hence students are very busy in the rat race to score marks and ranks and are not in a position to realize the main purpose of education i.e. to learn new things. School is not just a place where you learn a few subjects, pass the examinations and move out. It is a place where the foundation for the rest of the life like the moral values, behavior in the society and creativity are laid for tomorrow's generation. But the present education hardly concentrates on these issues. There are very few schools where moral science and personality development counseling are part of their curriculum. There are very few schools with a proper school ground and art room, aside from the importance given to the extracurricular activities.

$2^{228}/2^n=512$ . Find n? n=219

- 1)Furniture
- 2)plant
- 3)seed
- 4)three.    Correct form?? Ans: 3-->2-->1

odd one out?

- 1)PSQR
- 2)SVTU
- 3)MPON
- 4)DGEF

Ans: MPON

Which statement satisfies. Which one is greater?

Stmt1) B greater than A

Stmt2) D and E both are great than B

Stmt3) C less than B

Insufficient data. All the statements put together are not sufficient.

3 7 13 21 ? Ans: 31

GIVE people power and discretion, and whether they are grand viziers or border guards, some will use their position to enrich themselves. The problem can be big enough to hold back a country's development. One study has shown that bribes account for 8% of the total cost of running a business in Uganda. Another found that corruption boosted the price of hospital supplies in Buenos Aires by 15%. Paul Wolfowitz, the head of the World Bank, is devoting special efforts during his presidency there to a drive against corruption.

For most people in the world, though, the worry is not that corruption may slow down their country's GDP growth. It is that their daily lives are pervaded by endless hassles, big and small. And for all the evidence that some cultures suffer endemic corruption while others are relatively clean, attitudes towards corruption, and even the language describing bribery, is remarkably similar around the world.

In a testament to most people's basic decency, bribe-takers and bribe-payers have developed an elaborate theatre of dissimulation. This is not just to avoid detection. Even in countries where corruption is so common as to be unremarkable and unprosecutable—and even when the transaction happens far from snooping eyes—a bribe is almost always dressed up as some other kind of exchange. Though most of the world is plagued by corruption, even serial offenders try to conceal it.

Related items

- Corruption and the law: BarefacedDec 19th 2006

- Party funding: Looking worseDec 19th 2006

One manifestation of this is linguistic. Surprisingly few people say: “You are going to have to pay me if you want to get that done.” Instead, they use a wide variety of euphemisms. One type is quasi-official terminology. The first bribe paid by your correspondent, in Ukraine in 1998, went to two policemen so they would let him board a train leaving the country. On the train into Ukraine, the customs officer had absconded with a form that is needed again later to leave the country. The policemen at the station kindly explained that there was a shtraf, a “fine” that could be paid instead of producing the document. The policemen let him off with the minimum shtraf of 50 hryvnia (\$25).

Another term widely used at border crossings is “expediting fee”. For a euphemism it is surprisingly accurate: paying it will keep your bags, and perhaps your contraband, from being dumped onto a floor and sifted through at a leisurely pace. (A related term, used in India, is “speed money”: paying it can get essential business permits issued considerably faster.)

Paul Lewis, an analyst with the Economist Intelligence Unit (a sister company to The Economist), describes the quasi-business terminology typically used for bribery in the post-communist privatisations of eastern Europe. A mostly useless but well-connected insider at the company is hired as a “consultant”. The consultant is paid a large official “fee”, nominally for his industry expertise, on the understanding that he will cut in the minister and other decision-makers.

A second type of euphemism dresses up a dodgy payment as a friendly favour done by the bribe-payer. There is plenty of creative scope. Nigerian policemen are known to ask for “a little

something for the weekend”. A North African term is “un petit cadeau”, a little gift. Mexican traffic police will suggest that you buy them a refresco, a soft drink, as will Angolan and Mozambican petty officials, who call it a gazoso in Portuguese. A businessman in Iraq told Reuters that although corruption there is quite overt, officials still insist on being given a “good coffee”.

Double meaning can help soothe the awkwardness of bribe-paying. Baksheesh, originally a Persian word now found in many countries of the Middle East, can mean “tip”, “alms” and “bribe”. Swahili-speakers can take advantage of another ambiguous term. In Kenya a machine-gun-wielding guard suggested to a terrified Canadian aid worker: “Perhaps you would like to discuss this over tea?” The young Canadian was relieved: the difficulty could be resolved with some chai, which means both “tea” and “bribe”.

India lives in several centuries at the same time. Somehow we manage to progress and regress simultaneously. As a nation we age by pushing outward from the middle—adding a few centuries on either end of the extraordinary CV. We greater like the maturing head of a hammerhead shark with eyes looking in diametrically opposite directions.

I don’t mean to put a simplistic value judgment on this peculiar form of “progress” by suggesting that Modern is Good and Traditional is Bad—or vice versa. What’s hard to reconcile oneself to, both personally and politically, is the schizophrenic nature of it. That applies not just to the ancient/modern conundrum but to the utter illogic of what appears to be the current national enterprise. In the lane behind my house, every night I walk past road gangs of emaciated laborers digging a trench to lay fiber-optic cables to speed up our digital revolution. In the bitter winter cold, they work by the light of a few candles.

It’s as though the people of India have been rounded up and loaded onto two

convos of trucks (a huge big one and a tiny little one) that have set off resolutely in opposite directions. The tiny convoy is on its way to a glittering destination somewhere near the top of the world. The other convoy just melts into the darkness and disappears. A cursory survey that tallies the caste, class and religion of who gets to be on which convoy would make a good Lazy Person's concise Guide to t

### Questions

1. Why does the author calls 'progress' as peculiar?
  - a. Because Modern is good and traditional is bad.
  - b. **Because of its unbalanced nature.**
  - c. Because it differs politically and personally. D. None of these.
2. What do you infer from the sentence -'For some of us, life in .....but emotionally and intellectually'?
  - a. A person has one leg in one truck and the other in the second truck.
  - b. A person meets with an accident.
  - c. **The nation is moving in two different directions.**
  - d. The nation is suffering from many road accidents
3. How does the author feel about 'Globalisation' in India?
  - a. Curious b.**Hopeless** c. Enthusiastic d. **Speculative**
4. What does the sentence "We greater like the maturing head of a hammerhead shark with eyes looking in diametrically opposite directions.' implies?
  - a. Indian people are barbaric in nature.
  - b. **We are progressing in some areas and regressing in the others.**
  - c. India has a diverse culture.

- d. Some people are modern while the others are traditional in approach.
5. What do you infer from the sentence in context of the passage-'India lives in several centuries at the same time.'?
- We are progressing in some areas and regressing in the others.**
  - People from different countries are living in India.
  - India has a diverse culture.
  - Some people are modern while the others are traditional in approach.
6. What do you infer from the following lines-'In the lane behind my house, every night I walk past road gangs of emaciated labourers digging a trench to lay fiber-optic cables to speed up our digital revolution? In the bitter winter cold, they work by the light of a few candles.'?
- India has a balanced mixture of both traditional and modern people.
  - Progress is unbalanced.**
  - Digital revolution is very important for our economic growth.
  - There is shortage of electricity in India.
7. What does the phrase "cultural insult" imply?
- People from one culture do not respect people from the other cultures.
  - Disrespect of British towards Indian Culture.**
  - White people's definition for us. D. Ill-treatment at hands of British
8. Why does the response towards 'Globalisation in India' differs in different parts of India?
- Due to different literacy levels. B. Due to religious diversity in India.
  - It will not benefit all sections of the society.**
  - It may not have all the answers to India's current problems.

Bribery meaning--->**an attempt to make someone do something for you by giving the person money, presents, or something else that they want.**

Which expression gives maximum number of nodes at level I of binary tree (the root is at level I)

Ans:  $2^1 - 1$

Identify the Lowest level format to which computer converts a higher language program before execution

Ans: Compiler

A programmer is making a database of animals in zoo along with their properties possible animals dog lion and zebra is attributes herbivorous colour and not eternal the programmer is an object oriented programming faradism for this how is the system by conceptualized

The system must be designed in the following way - class: Animal; objects: dog, lion and zebra; data members: is Herbivorous, colour and is Nocturnal. The Object Oriented Programming is referred as the model of programming language that is organised around the objects rather than the "actions" and data than the logic. A program is generally viewed as the logical procedure which takes the input data and produces output after processing it.

sorting algorithm iteratively traverses through a list to exchange the first element with any element less than it. It then

repeats with a new first element. What is this sorting algorithm called?

Ans: Selection Sort

X and Y are asked to write a program to sum the rows of a 2x2 matrix stored in an array A.

X writes the code (Code A) as follows:

for n = 0 to 1

sumRow1[n] = A[n][0] + A[n][1]

end

Y writes the code (Code B) as follows:

sum Row 1[0] = A[0][0] + A[0][1]

sum Row1[1] = A[1][0] + A[1][1]

Which of the following statements is correct about these codes if no loop unrolling is done by the compiler?

Code A would execute faster than Code B.

Code B would execute faster than Code A.

Code A is logically incorrect

Code B is logically incorrect

Ans: B is faster than A.

A language has 28 different letters in total. each word in the language is composed of maximum 7 letters. you want to create a data-type to store a word of this language. you decide to store the word as an array of letters. how many bits will you assign to the data-type to be able to store all kinds of words of the language.

Ans: 35 bits

Reversing second half of array elements code

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    int *a,n,i,j,temp;
```

```
    printf("Enter size of array:");
```

```
    scanf("%d",&n);
```

```

a=malloc(sizeof(int)*n);

printf("Enter %d Elements:",n);

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

{

    scanf("%d",&a[i]);

}

for(i=n/2,j=n-1;i<j;i++,j--)

{

    temp=a[i];

    a[i]=a[j];

    a[j]=temp;

}

printf("After reversing the second half:\n");

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

{

    printf("%d ",a[i]);

}

return 0;
}

```

Sorting is not possible using

Ans: Deletion

Why run time considered more

compile time may vary programmer to programmer according to their program size for same task but run time only vary according to algorithm, thus run time matters.

What happens if some indentation happens in c++

Ans: Nothing

What expression gives maximum number of nodes

Ans:  $2^l$  where l is level

What is the largest power of 20 obtained in 100 factorial

Ans: 24

How to Balance between professional and personal life.

The topic “Work-Life Balance” has been selected because with increases industrialization and urbanization since Second World War, women labor force participation has increased tremendously and we observe many dual earner couples. These families used to count on help from non-working women relatives. However, traditional family care support has also decreased with time due to a host of reasons (less family members are available to look after their parents in old age due to increased rural-urban migration and increased need for income).

The current situation (where workers have to look after their children as well as work/earn) leads to a situation called work-family conflict. It refers to pressures from work and family domains, which are incompatible with each other in such a way that meeting requirements of one role makes it difficult to satisfactorily fulfill the other role. According to research, working long hours is one of the sources of work-family conflict. Hence, to maintain the balance between work and family life, the theory “work-life balance” has been adopted. According to which the growth of any company depends on the interrelated performance and commitment of the employees. Both personal and professional lives are interrelated and difficult to be separated from each other. Organization needs to keep its employees happy and satisfied to get the maximum productivity . The balance is achieved when an individual’s rights are fulfilled by the people from whom he/she is expecting to sort out the core problems and issues regarding professional and personal life. Previous studies elaborates that employees satisfaction, motivation, profitability, productivity, recruitment and retention policies can be improved by adopting flexible working arrangements and full determination towards the betterment of the employees and organizational goals.

Today, the work life balance is a common issue at every workplace. The work life balance creates balance between the work and life includes career and the life style. It is the state of equilibrium where any person is able to get satisfies with personal and professional life. Balancing time with parents, spouse, children, friends, relatives, colleagues, supervisors and subordinates is the biggest challenge to be achieved. The empirical study gives the idea of work life balance performed in past. Work life balance has become a challenge throughout the world. The study will identify the assessment of employees about their organization from the work life balance prospective and how it is practiced at their workplace. This will also measure the level of job satisfaction of employees of the PIA in relation to their socio-economic factors. It is to be cited from , the individuals are facing the challenges that how they balance their work life and personal life. It is to be cited from that this study is also supported by (lewis et al, 2012) who study younger's in four European countries and found the positive desire related to work life balance. Without balance in working life and personal life, it is very difficult to maintain and control the personal and working life. In order to be satisfied, the individual must create the balance between their work life and personal life. It is found generally that work life balance is too difficult.

We want to become technically\_\_\_\_interns of technological results

Ans: recognized

**The monk wanders here and there in search of silence and peace his lives a \_\_\_\_ life**

**Ans: Simple/peaceful**

A writer always is having his own vision of life.

Select an alternative to sentence options:

A. Always has

B. Always had

**C.always have**

D. No improvement needed

24:50::102: Ans: 206

Which of the following sorting algorithms yield approximately the same worst-case and average case running time behaviour in  $O(n \log n)$ ?

A)Bubble sort and selection sort

**B)Heap sort and merge sort**

C)Quick Sort and radix sort

D)Tree sort and median-of-3-quicksort

4. which sorting algorithm traverses through a list?

Ans: bubble sort

Which of the following statement is not true about breadth -first search (BFS) in an undirected graph starting at a vertex v?

a. BFS identifies all vertices reachable from v.

b. using an adjacecy list instead of anadjacecy matrix can improve the wrost case complexity to on  $(n+m)$ .

c. BFS can't be used to check for a cycle in the graphs.

d. BFS can be used to identify the furthest vertex from V in any graph.

Ans: 1-->true 2-->false 3-->false 4-->true

What best describes space complexity of a program ?

Space complexity is a measure of the amount of working storage an algorithm needs. That means how much memory, in the worst case, is needed at any point in the algorithm. As with time complexity, we're mostly concerned with how the space needs to grow, in big-Oh terms, as the size N of the input problem grows.

What is the least perfect square that is divided by 24,60,30?ANS:3600

Life in small town and big city essay

Many people who live in a big city abandon there and prefer to live in small towns these days. They get bored with rush city life. They want to live a relaxed life. When they live in a big city, they have big problems with people's health and money affairs. However, when they live in a small town they go far away from health problems, traffic and expensive life for these reasons many people prefer to live in a small town.

First of all, there is a lot of environmental pollution in a big city and they cause illnesses such as asthma. People who are in small-town expose less to these problems because generally there aren't factories, many cars and big construction in there. Also, a small town has many green places. People know now that their life is valuable and they should spend it well. They want to live a quality life outside of stress or some health problem. According to research, people who live in a green place have a longer life than other people.

Furthermore, there are many kinds of products in a big city but generally, they are expensive because when these products arrive there, fuel oil money and employees' salary add to their cost. For that, vegetables and fruits cost very high in a big city. These events don't exist in a small town because they produce their vegetables and fruits. They can consume them or sell them in a villager bazaar. Also, house rent is very high especially if it is close to the city center. On the other hand, there is no problem with the small town because the population is low and there isn't rivalry there for that reason many people prefer a small town.

To summarize, people who want a calm, healthy and cheap life should live in a small town. If they do that, maybe they spend their life in a good situation.

A data type is stored as an 6 bit signed integer. Which of the following cannot be represented by this data type?

Ans: 18

Ravi and Rupali are asked to write a program to sum the rows of a 2X2 matrices stored in the array A.

Ravi writes the following code (Code A):

```
for n = 0 to 1 sumRow1[n] =
```

$A[n][1] + A[n][2]$  end

Rupali writes the following code (Code B):

$sumRow1[0] = A[0][1] + A[0][2]$

$sumRow1[1] = A[1][1] + A[1][2]$

Comment upon these codes (Assume no loop-unrolling done by compiler):

Ans: Code B would execute faster than code A.

Abhinav wants to find the largest number in a given list of 20 numbers. Which of the

following is an efficient approach to do this?

Op 1: Use bubble sort to sort the list in descending order and then print the first number of the series.

Op 2: Use selection sort to sort the list in descending order and then print the first number of the series.

**Op 3: Implement one iteration of selection sort for descending order and print the first number**

**in the series.**

Op 4: None of these

Consider an array on which bubble sort is used. To which of the given elements will the bubble sort compare the element  $A[x]$  with, in a single iteration.

a.  $A[x+1]$

b.  $A[x+2]$

C.  $A[x+2x]$

d. All the above

Sharmili wants to make a program to print the sum of all perfect cubes, where the

value of the cubes go from 0 to 100. She writes the following program:

```
integer i = 0, a // statement 1  
  
integer sum = 0; a = ( i * i * i )  
  
while ( i < 100 ) // statement 2  
  
{  
  
    sum = sum + a // statement 3 i  
  
    = i + 1 a = ( i * i * i ) //  
  
    statement 4  
  
}  
  
print sum
```

Does this program have an error? If yes, which one statement will you modify to correct the program?

Op 1: Statement 1

**Op 2: Statement 2**

OP 3:statement 3

Op4:statement 4

Op5: no error

What will be the output of the following pseudo-code statements:

```
integer a = 456, b, c, d = 10 b = a/d c = a - b print c
```

Ans: 411

A librarian has to rearrange the library books on a shelf in a proper order at the end of each day. Which of the following sorting techniques should be the librarian's ideal choice?

Ans: Insertion sort

When is a **Constructor** called?

Each time an object is created using a new() keyword, at least one **constructor** (it could be the default **constructor**) is invoked to assign initial values to the data members of the same class. A **constructor is invoked** at the time of object or instance creation.

Vexing meaning

Ans: causing annoyance, frustration, or worry.

**The temperature in Michigan dipped so low that**

**Ans: lake michigan have dropped more than 20 degree in 24 hours**

**A programmer writes a code snippets in which set of 3 lines occurs 10 times in different pathway programs what program concepts should be used to shorten code length.**

**Ans: The programming concept that the programmer should use here is *modularity or looping*.**

**Of all the fitness and wellness activities customary in india, artistic yoga is the new kid in town. it has successfully earned a pat on the back from whosoever has lent an ear to the latest advancements. artistic yoga combines the suaveness of yoga and the frenzy of modern cardio-vascular exercises. the technique involves performance of various aasanas and pranayams followed by walking on treadmill, stair climbing, cycling and so on. the activities are performed in a cyclic order and the aasana or pranayam that is done in the beginning is repeated in the end.**

**Ans: all the activities formed at the beginning of artistc yoga and also repeated at the end**

**The unique Iron Age Experimental Centre at Lejre, about 40 km west of Copenhagen, serves as a museum, a classroom and a place to get away from it all. How did people live during the Iron Age? How did they support themselves? What did they eat and how did they cultivate the land? These and a myriad of other questions prodded the pioneers of the Lejre experiment.**

Living in the open and working 10 hours a day, volunteers from all over Scandinavia led by 30 experts, built the first village in the ancient encampment in a matter of months. The house walls were of clay, the roofs of hay - all based on original designs. Then came the second stage - getting back to the basics of living. Families were invited to stay in the 'prehistoric village' for a week or two at a time and rough it Iron Age-style.

Initially, this experiment proved none too easy for modern Danes accustomed to central heating, but it convinced the centre that there was something to the Lejre project. Little by little, the modern Iron Agers learnt that their huts were, after all, habitable. The problems were numerous - smoke belching out from the rough-and-ready fireplaces into the rooms and so on. These problems, however, have led to some discoveries: domed smoke ovens made of clay, for example, give out more heat and consume less fuel than an open fire, and when correctly stoked, they are practically smokeless.

By contacting other museums, the Lejre team has been able to reconstruct ancient weaving looms and pottery kilns. Iron Age dyeing techniques, using local natural vegetation, have also been revived, as have ancient baking and cooking methods.

**A) What is the main purpose of building the Iron Age experimental center?**

- a) Prehistoric village where people can stay for a week or two to get away from modern living**
- b) Replicate the Iron Age to get a better understanding of the time and people of that era**
- c) To discover the differences between a doomed smoke oven and an open fire to identify the more efficient of the two**
- d) Revive activities of ancient women such as weaving, pottery, dyeing, cooking and baking**

**Solution: Option b**

**B) What is the meaning of the sentence "Initially, this experiment proved none too easy for modern Danes accustomed to central heating, but it convinced the centre that there was something to the Lejre project."?**

- a) Even though staying in the huts wasn't easy for the modern people, the centre saw merit in the simple living within huts compared to expensive apartments**
- b) Staying in the huts was quite easy for the modern people and the centre also saw merit in the simple living within huts compared to expensive apartments**
- c) The way of living of the Iron Age proved difficult for the people of the modern age who are used to living in luxury**
- d) The way of living of the Iron Age proved very easy for the people of the modern age since it was hot inside the huts, and they were anyway used to heated rooms**

**Solution: Option c**

- C) What can be the title of the passage?**
- a) Modern techniques find their way into pre-historic villages**
  - b) Co-existence of ancient and modern times**
  - c) Glad to be living in the 21st Century**

**d) Turning back time**

**Solution: Option d**

**D) From the passage what can be inferred to be the centre's initial outlook towards the Lejre project?**

**a) It initiated the project**

**b) It eagerly supported it**

**c) It felt the project was very unique**

**d) It was apprehensive about it**

**Solution: Option a**

**AUTOMATA FIX:**

**Q) 1**

1 1

1 1 1

```
void patternPrint(int num)
```

```
{
```

```
int print=1,i,j;
```

```
for(i=0;i<num;i++)
```

```
{
```

```
for(j=0;j<=i;j++);
```

```
{
```

```
printf("%d ",print);
```

```
}
```

```
printf("\n");
```

```
}
```

```
}
```

**Ans: semicolon after for loop**

**Q) return true if given date is july 5th**

.

```
int checkBirthDay(char* month, int day)
```

```
{
```

```
if((strcmp(month, "July")) || (day == 5))
```

```
return 1;
```

```
else
```

```
return 0;
```

```
}
```

**Ans: both conditions has to be satisfied. so and operator has to be**

**applied.**

**strcmp return 0 when equal. so check 0 or not**

```
int checkBirthDay(char* month, int day)
```

```
{
```

```
if(strcmp(month, "July") == 0) && (day == 5))
```

```
return 1;
```

```
else
```

```
return 0;
```

```
}
```

**Q) find x power y, given function that computes positive power only.**

```
float allExponent(int baseValue,int exponentValue){
```

```
float res=1;
```

```
if(exponentValue >=0)
```

```
{
```

```
res=(float) positiveExponent(baseValue,exponentValue);
```

```
}
```

```

else
{
//write your code here for negative value of exponent

}

return res;
}

```

**Ans:  $1/x$  equals to  $x^{-1}$ .**

**Q) check whether given points form a triangle or not**

```

int isTriangle(Point *P1, Point *P2, Point *P3)

{
//write your code here
}

```

**Ans: area of a triangle given three points is**

$(1/2) * (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2))$ .

**if this is zero then it does not form a triangle**

```

int isTriangle(Point *P1, Point *P2, Point *P3)

{
int a = P1->x * (P2->y - P3->y) + P2->x * (P3->y - P1->y) +
P3->x * (P1->y - P2->y);

if (a == 0)

return 0;
}

```

```
else
```

```
    return 1;
```

```
}
```

### Q) sorting problem

```
void selectionSortArray(int len,int *arr)
```

```
{
```

```
    int x=0,y=0;
```

```
    for(x=0;x<len;x++){
```

```
        int index_of_min=x;
```

```
        for(y=x;y<len;y++){
```

```
            if(arr[index_of_min]>arr[y]){
```

```
                index_of_min=y;
```

```
}
```

```
}
```

```
    int temp=arr[x];
```

```
    arr[x]=arr[index_of_min];
```

```
    arr[index_of_min]=temp;
```

```
}
```

```
}
```

**Ans: index\_of\_min element should be compared with y not with x**

### Q) program to perform descending order sorting

```

void descendingSortArray(int len,int *arr)

{
    int small, pos, i, j, temp;

    for(i=0;i<=len-1;i++){

        for(j=i;j<len;j++){

            temp=0;

            if(arr[i]>arr[j]){

                temp=arr[i];

                arr[i]=arr[j];

                arr[j]=temp;

            }

        }

    }

}

```

**Ans: given program performs ascending order sorting.**

**reverse the condition to get descending order.**

On Thursday night, a crowd gathered at Union Square in Manhattan for a fond and spontaneous memorial to Michael Jackson. A few hundred onlookers formed a circle, leaving enough space in the middle for the grand standers and the brave to dance like the King of Pop', or try to Watching this spectacle, you had to wonder: When will this happen again? When will another pop culture figure mean so much to so many that people are moved to assemble, hug and dance?

Michael Jackson has sold an estimated 100 million copies worldwide of the 1982 album \* Thriller", that spent more than 31 weeks at the top of the Billboard charts. It's one of those high-water marks that nobody will touch because record stores are vanishing, and along with them, mega-hit albums are vanishing too. And it's rare for an album to last even three weeks at the top

When the Beatles were on "The Ed Sullivan Show" in 1964, more than 70 million people watched it, that is more than one third of the entire population of the United States, Yes, the Beatles were that good. But at that time, there were three networks and the radio. No online messaging apps, video games, movie multiplexes, malls or a dozen other potential drains on an audience. There weren't a lot of rock bands, either. And now that anyone with a computer has a miniature studio, and anyone with an internet connection can post a song, there are more dienreboentes, and artists than ever.

sentence mistake

**Correct sentence:**

**1- On Thursday night, a crowd gathered at Union Square in Manhattan for a fond and spontaneous memorial to Michael Jackson.**

**2- A few hundred onlookers formed a circle, leaving enough space in the middle for the grandstanders and the brave to dance like the King of Pop. or try to.**

**4- Michael Jackson has sold an estimated 100 million copies worldwide of the 1982 album "Thriller", that spent more than 31 weeks at the top of the Billboard charts. It's one of those high-water marks that nobody will touch because record stores are vanishing, and along with them, mega-hit albums are vanishing, too.**

**5- And it's rare for an album to last even three weeks at the top.**

**6- When the Beatles were on "The Ed Sullivan Show" in 1964, more than 70 million people watched it, that is, more than one-third of the entire population of the United States.**

**What is the function contained within a class is called?**

**Ans: Member function**

**10 7 10 12 14**

**Ans:13**

**Lexical meaning**

**Ans: relating to the words or vocabulary of a language.**

**One day raja left home and cycled 5km southwards then turned left and cycled 3km and again turned left and cycled 2km. How kms to straight to raja home.**

**Ans: 7.28 km**

**If mohan is brother of rohan's grandmother then how is rohan related to mohan**

**Ans: Grand child**

**If gulmohar is coded as tfonlszi then pipal is coded as**

**Ans: krkzo**

**What is the least no. By which 16800 is divided and is perfect square root**

**Ans:159**

**Value of  $(1/512)^{1/9}$**

**Ans:  $\frac{1}{2}$**

**If k is integer divided by 2 5 and 13 then what is the nxt integer divided by these 3**

**K+130 or 260**

**Computer is written with this code GKQLYPIN the Senate is written in which code?**

**Ans: WARWXA**

**The number 456'85 is completely divisible by 3. smallest whole digit number in place of can be:**

**Ans: 2**

**Shobhit bought 300 litres of milk at rs. 19 per litre. He added 200 litres of water to it and sold 400 litres of this milk at rs. 20 per litre. To the rest, he added 10 litres more water and then sold it for rs. 15 per litre. If he used mineral water that costs rs.10 per litre, then the total money earned by shobhit is:**

**Ans: 1850**

**-3.4 is a number on real number line if we subtract 1.**

**Ans: -4.4**

**X walks 6 km towards east from a point a and from the same point a, y walks 8 km towards south. How far are the two friends from each other now?**

**Ans: 10km**

**What is sum of the 2 consecutive number. The difference of whose square is 19**

**Ans: 19**

**what implied by the argument of a function**

**Ans: when a function is called, some value is passed by the calling function, known as the argument.**

**If sound is coded as rntmc then work is coded as...?**

**Ans: vnqj**

**fill the missing number in the sequence 2,3,7,8,13,14 \_\_\_\_\_**

**Ans: 20**

**If the square of one number is 25 and LCM and HCF of those two numbers are 5 and 35 respectively, then what is the other number?**

**Ans: 35**

**What is the difference between LCM and HCF of 20, 30, 40?**

**Ans: 110**

**Correct the sentence incase of any error : She has been lived in Michigan since 2020.**

**Ans: She has been living in Michigan since she was eight.**

**Which of the following is a stable sorting method?**

**Ans: Merge Sort**

How does inheritance relate to abstraction?

1. A base class is an abstraction of all its derived classes.
2. **A derived class is an abstraction of all its base classes.**
3. Base and derived classes are abstractions for each other.
4. Inheritance prevents abstraction

```
int main() {
```

```
    integer a=984,b=10;
```

```
    float c=a/b;
```

```
    printf(c);
```

```
}
```

What is the output of the above code?

Ans: 98.000000

A machine worth 1,80,000 decreases 18% per annum. What is the worth of it after 18 months?

Ans: 134,316

What is the highest power of 2 in:  $1800 \times 25 \times 21 \times 4^8 \times 45^2$

Ans: 6

Nature provides us with much but when we abuse nature we risk disaster write response explaining how we harm ourselves when we harm environment. Essay

Natural capital is everything nature provides us for free. It is what our economy is built upon. We add man-made capital in the shape of houses, factories, offices and physical infrastructure, and human capital with our skills, ideas and science.

Natural capital should, therefore, be at the heart of economics and economic policy – but it isn’t. As a consequence we abuse nature, drive species to extinction, and destroy ecosystems and habitats without much thought to the consequences. The damage won’t go away; as we wipe out perhaps half the species on the planet this century and induce significant climate change, the economic growth we take for granted will be seriously impaired. Put simply, our disregard for natural capital is unsustainable – it will not be sustained.

Just as we try to maintain and enhance our own assets – our houses, cars and our knowledge and skills – so too should the broader economy avoid running down its base of natural capital assets. Some natural assets will be used up – such as the non-renewables like North Sea oil and gas – but even here we should be mindful of compensating future generations for what we will not therefore bequeath to them.

The natural assets that really matter are the renewables – the ones nature keeps on providing us for free, forever – provided we don’t deplete them beyond the threshold of sustainable reproduction. We can for example carry on for hundreds of thousands of years harvesting herring from the North Sea, as long as we do not overfish them. The potential value of all those fish forever is enormous.

For the economy as a whole, national accounts need to provide for this capital maintenance as a first call on its revenue. If they did, and if we had a proper balance sheet, then the economic growth declared would be the sustainable one. It would be lower, too. Put another way, by not maintaining our assets, we are living beyond our means – let alone also borrowing from the future to finance the deficit, and making future generations pay for our excessive current consumption.

A sustainable level of economic growth – driven by all the new technologies that keep coming along – would be lower, but we would nevertheless be better off in the long run. We would not

then face the consequences of the loss of so much of our natural environment and climate change.

Unlike the climate change problem, natural capital has a big spatial dimension. Climate change is all about a small number of gases in the atmosphere and it does not matter where they are emitted. Natural capital comes at every level, and location matters. It includes the great global biodiversity hotspots, like the Amazon, and it includes everyone's backyard and every park and garden. Everyone can therefore make a difference to natural capital in ways that they cannot to the climate change problem.

Think of some of the things you could do today. If your front garden is paved and concreted over, you could break it up and allow wild flowers to flourish, which in turn would help the bees. So barren is much intensive agricultural land, as it is sprayed with pesticides and herbicides and its soils supplemented with nitrogen fertilisers, that bees often find cities much better habitats than the countryside. You could put away the slug pellets and the weed killer.

But the really big gains come from three policy changes, all of which have an impact on the choices we make. The first is compensation: the idea that if damage is done to our natural capital, there must be compensating increases elsewhere. The second is pollution taxes, paying for the damage caused from carbon, pesticides and other emissions. The third is a nature fund, created from the running down of the non-renewables like North Sea oil and gas.

As Britain faces a population increase of around 10 million in the next couple of decades, and as all the main political parties are committed to building more than 200,000 houses per year, the HS2 high-speed railway which will initially link London and Birmingham, and more airport runways, there is little doubt that the natural environment faces challenges matched only by the devastation caused by the intensification of agriculture encouraged by the Common Agricultural Policy.

Natural capital is in for a shock, unless compensating improvements are made. These could be really big – improvements in complete river catchments like the Thames and new marine protected areas. They can be small and local too; every urban citizen could be guaranteed access

to a green space within 500m of where they live, addressing not only the space children need to play outdoors, but also the poor air quality that blights our cities. These are parts of the prize of taking natural capital seriously.

what sum of money will accumulate to rs.5,300 at 8% simple interest in 9 months?

Ans: 5,000 Rs

A number X when divided by 13 leaves remainder 12 what is the remainder when we divided  $X^{13}$  by 13?

Ans: 12

An employee has to attain 70% marks during appraisals to get a promotion. He gets 75 marks and doesn't qualify for a promotion since he falls short of 30 marks in the ratings. What is the maximum score that an employee can attain in the appraisal ratings?

Ans: 150

This new technology has the potential to provide handsome returns even though it is at a \_\_\_\_\_ stage in India.

Ans: Nascent

Which data structure may produce an overflow error even though the current number of elements in it lower than its size

Ans: Queue

As soon as I turn the ignition key, the engine caught fire

Ans: As soon as I turn the ignition key, the engine caught fire.allturn an ignition key

A \_\_\_\_\_ person is bound to insure himself against all types of losses.

Ans: Circumspect

A function in the base class is redefined in inherited class. Which method describes this situation?

Overloading

Which of the given options implies that there are two

loops that are nested?

1 Two loops, one after the other

**2 Two loops, one inside the other**

3 One loop with two different iteration counts

4 Two loops with the same iteration count

The market was full of people ..... attractive clothes of different styles.

Ans: wearing

Sort array in c:

```
void sortArray(int len, int *arr)
```

```
{
```

```
    int i, max, location, temp, j, k;
```

```
    if(len%2 == 0)
```

```
{
```

```
    for(i=0;i<len;i++)
```

```
{
```

```
        max=arr[i];
```

```
        location = i;
```

```
        for(j=i;j<len;j++)
```

```
if(max>arr[j])  
{  
    max=arr[j];  
    location = j;  
  
}  
  
temp=arr[i];  
  
arr[i]=arr[location];  
  
arr[location]=temp;  
  
}  
  
}  
  
else  
  
{  
    for(i=0;i<len;i++)  
    {  
        max=arr[i];  
        location = i;  
        for(j=i;j<len;j++)  
        if(max<arr[j])  
        {  
            max=arr[j];  
            location = j;  
        }  
    }  
}
```

```
    }

    temp=arr[i];

    arr[i]=arr[location];

    arr[location]=temp;

}

}

}
```

Count digits in c

```
#include <stdio.h>

int main() {

    long long n;

    int count = 0;

    printf("Enter an integer: ");

    scanf("%lld", &n);

    // iterate until n becomes 0

    // remove last digit from n in each iteration

    // increase count by 1 in each iteration

    while (n != 0) {

        n /= 10;    // n = n/10
```

```

++count;

}

printf("Number of digits: %d", count);

}

```

Binary search comparision

**Selection sort in descending order in c:**

```

#include <stdio.h>

void main ()
{

    int number[30];

    int i, j, a, n;

    printf("Enter the value of N\n");
    scanf("%d", &n);

    printf("Enter the numbers \n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);

    /* sorting begins ... */

```

```
for (i = 0; i < n; ++i)
{
    for (j = i + 1; j < n; ++j)
    {
        if (number[i] < number[j])
        {
            a = number[i];
            number[i] = number[j];
            number[j] = a;
        }
    }
}
```

```
printf("The numbers arranged in descending order are given below\n");
```

```
for (i = 0; i < n; ++i)
{
    printf("%d\n", number[i]);
}
```

```
}
```

Binary search in c:

```
#include <stdio.h>
int main()
{
    int c, first, last, middle, n, search, array[100];
```

```
printf("Enter number of elements\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

scanf("%d", &array[c]);

printf("Enter value to find\n");

scanf("%d", &search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while (first <= last) {

if (array[middle] < search)

first = middle + 1;

else if (array[middle] == search) {

printf("%d found at location %d.\n", search, middle+1);

break;
}
```

```

    }

    else

        last = middle - 1;

        middle = (first + last)/2;

    }

    if (first > last)

        printf("Not found! %d isn't present in the list.\n", search);

    return 0;

}

```

What does the following function do? Function operation (int a, int b) {  
if (a < b)  
{return operation (b, a)  
}  
else {  
return a} }

Ans : Returns the max of (a,b)

A programmer is making a database of animals in a zoo and their properties. The possible animals are dog, lion and zebra. Each one has as attributes isHerbivorous, colour and isNocturnal. She uses the object oriented programming paradigm for this. How will she conceptualize the system?

Ans : The system must be designed in the following way - class: Animal; objects: dog, lion and zebra; data members: is Herbivorous, colour and is Nocturnal.

Choose the correct answer. A pseudo-code which is similar to that of C++ and self-explanatory  
An accessible member function or data member for an object are accessed by the statement  
objectname.functionname or objectname. data member name respectively. class brush

```
{ Private: integer size, colorcode  
function getdata( ) {—}//Statement 1 public: integer name // Statement 2  
function putdata(){...} }  
function main { brush b1, b2  
print b1.name //Statement 3  
b2.getdata() //Statement 4 }
```

Deleting which line will correct the error in the code?

Ans : Statement 4

A sorting algorithm traverses through a list, comparing adjacent elements and switching them under certain conditions. What is this sorting algorithm called?

Ans : Quick Sort

A sorting algorithm iteratively traverses through a list to exchange the first element with any element less than it. It then repeats with a new first element. What is this sorting algorithm called?

Ans : Selection sort

What will be the output of the following pseudo-code statements:

```
integer a = 456, b, c, d = 10 b = a/d c = a - b print c
```

Ans : 411

A language has 28 different letters in total. Each word in the language is composed of maximum 7 letters. You want to create a data-type to store a word of this language.

You decide to store the word as an array of letters. How many bits will you assign to the data-type to be able to store all kinds of words of the language.

Ans : 35 bits

Which of the following sorting method is stable ?

Ans : Binary Insertion Sort

A programmer writes a program to find an element in the array A[5] with the elements: 8 30 40 45 70. The program is run to find a number “X”, which is found in the first iteration of binary search. What is the value of “X”?

Ans : 40

A developer writes the program given below to print the sum of the squares of the first five whole numbers (0 .. 4). Is the program correct? If not, which statement should be modified to correct the program?

```
integer i = 0 // Statement 1  
integer sum = 0 // Statement 2  
while (i<5) // Statement 3  
{
```

```
sum = i*i // Statement 4
```

```
i = i + 1 // Statement 5
```

```
}
```

```
print sum // Statement 6
```

Ans : Statement 4

x and y are asked to write a program to sum the rows of a 2x2 matrix stored in an array a.

x writes the code (code a) as follows:

```
for n = 0 to 1
```

```
sum row1[n] = a[n][0] + a[n][1]
```

```
end
```

y writes the code (code b) as follows:

```
sum row 1[0] = a[0][0] + a[0][1]
```

```
sum row 1[1] = a[1][0] + a[1][1]
```

of the following statements is correct about these codes if no loop unrolling is done by the compiler?

Ans : Code B would execute faster than Code A.

Essay - global Nuclear power

We are witnessing and harnessing the growth use of nuclear technology for power production and other applications from both developed and developing countries (IAEA, 2011). Regrettably according to International Atomic Energy Agency (IAEA), more than 100 member States who use radioactive materials that can be used to make dirty bombs are characterized by unsatisfactory control and management system (National Research Council, 2007). It is clear that global security challenges like illicit cross-border trafficking in arms, illegal immigrants, drugs, radiological, chemical and biological weapons which are global threats to international peace and security posed by armed conflict, terrorism, weapons proliferation and transnational organized crime groups cannot be managed by a single country (United Nations, 2016, IFPA, 2010). This is why nuclear and other radioactive materials are required by the IAEA for member States to have a tough alternative protection with effective capabilities to spot and capture their illegal movement both at borders and within their States (IAEA, 2007). Regardless of these international requirements, the porous borders and limited security resources has been critical challenge in developing countries which has left number of nuclear and other radioactive materials out of regulatory control.

The community engagement in security has been emphasized in fight against local and global crimes, for instance community has been engaged in fight against extremist groups, to improve illegal immigration, drug abuse, and other community security challenges. Proper community engagement in radioactive materials security across the borders and within States with porous borders and limited security resources can help to improve the response of enforcement agents to illicit and other cross border crimes.

Which sorts has avg and worst case as  $O(n\log n)$ ?

Ans : Heap Sort

```

Pattern
a
ab
abc
#include<stdio.h>
int main()
{
    int row,j;
    for (row='A'; row<='C'; row++)
    {
        for (j='A'; j<=row; j++)
        {
            printf("%c", j);
        }
        printf("\n");
    }
    return 0;
}

```

Difference in time calculation pgm complete

```

#include <stdio.h>
struct TIME {
    int seconds;
    int minutes;
    int hours;
};

void differenceBetweenTimePeriod(struct TIME t1,
                                 struct TIME t2,
                                 struct TIME *diff);

int main() {
    struct TIME startTime, stopTime, diff;

    printf("Enter the start time. \n");
    printf("Enter hours, minutes and seconds: ");
    scanf("%d %d %d", &startTime.hours,

```

```

    &startTime.minutes,
    &startTime.seconds);

printf("Enter the stop time. \n");
printf("Enter hours, minutes and seconds: ");
scanf("%d %d %d", &stopTime.hours,
      &stopTime.minutes,
      &stopTime.seconds);

// Difference between start and stop time
differenceBetweenTimePeriod(startTime, stopTime, &diff);
printf("\nTime Difference: %d:%d:%d - ", startTime.hours,
      startTime.minutes,
      startTime.seconds);
printf("%d:%d:%d ", stopTime.hours,
      stopTime.minutes,
      stopTime.seconds);
printf("= %d:%d:%d\n", diff.hours,
      diff.minutes,
      diff.seconds);
return 0;
}

// Computes difference between time periods
void differenceBetweenTimePeriod(struct TIME start,
                                  struct TIME stop,
                                  struct TIME *diff) {
    while (stop.seconds > start.seconds) {
        --start.minutes;
        start.seconds += 60;
    }
    diff->seconds = start.seconds - stop.seconds;
    while (stop.minutes > start.minutes) {
        --start.hours;
        start.minutes += 60;
    }
    diff->minutes = start.minutes - stop.minutes;
    diff->hours = start.hours - stop.hours;
}

```

Technology has become a new addiction. we are becoming slave to our own creation

In our quest to discover unknown things and to invent new things we have reached a stage where it is unimaginable and impossible to live without technology. Technology has been gradually taking over our lives and we are not even aware of the fact that we have become slaves to technology. We have become too dependent and reliant on technology that if it were to be taken away from us we would be confused and lost.

Technology is helping us in many ways like financial transaction, buying food and stuffs, paying our bills and booking our flight tickets in the comfort of our home. Similarly we can online education and degrees, instant medical advice on health issues, watching our favorite movies and many more. But although technology has been helping us, it has been pulling us away from the real world.

Human beings are social beings. We love the latest technological gadgets which help people to connect and communicate with each other. But in the process we have lost authentic relationships. Technology does not allow us to work as a team of real humans with real connection.

Psychologically, the silence without real human social life is making us more isolated and lonely. Depression is increasing but we don't like to blame technology because it has become a way of life for us.

Technology has stolen our physical well-being. We have less physical activities and many of our body parts are becoming less inactive and shutting down. New technologies can empower us to connect with people anywhere in the world. But the harsh reality is that these technologies have divided us. Within the last decade we have steadily and gradually become slaves to technology, we have become actors in our own story.

```
int a=984,b=10;float c; c=a/b;
```

output in c

Ans : 98

Saloni writes the code for a function that takes as input  $n$ , an even integer and calculates the sum of first  $n$  even natural numbers.

```
function sum( n )
```

```
{
```

```
if(n equals 2)
```

```
return 2
```

```
else
```

```
return (n + sum(n-2))
```

```
end
```

```
}
```

She then calls the function by the statement,  $\text{sum}(30)$ . How many times will the function  $\text{sum}$  be called to compute this sum?

Question

- i) 1 ii) 30 iii) 15 iv) 16

Ans 16

In the following sorting procedures, which one will be the slowest for any given array? (a)Quick sort (b)Heap sort (c)Merge Sort (d)Bubble sort

Heap sort

Pankaj and Mythili were both asked to write the code to evaluate the following expression:

$$a - b + c/(a-b) + (a-b)^2$$

Pankaj writes the following code statements (Code A):

```
print (a-b) + c/(a-b) + (a-b)*(a-b)
```

Ans : Code A uses lesser memory and it is slower than code B.

The tight curriculum of our education system leaves no room for imagination and creativity. Essay

I whole heartedly agree that the tight curriculum of our current education system leaves no room for imagination and creativity. The academic pressure is so much on them, they have forgotten their innate creativity and imagination. They are always under the pressure of studies. This pressure is put on them from the beginning of the session. Formative assessment activities of all the subjects, studies and their assignments of all the subjects, unit tests, co-curricular activities, homework, classwork, projects, practical exams, summative examination, date sheet, result, again the new term, etc. So much of pressure on the young growing children! Are we educating them or training them the art of being stressful throughout their childhood?

Even while studying inside class, they feel so burdened; they have come to dislike education. as a result there is hardly any creativity left in our country. Every year all the illustrious prizes for creativity are won by non-Indians or Indians living in advanced countries. Our educationists must rethink and revise the curriculum to encourage creativity and imagination in children --- not rot learning.

3,7,13,21... what is the Next series number

Ans : 31

Waxing means

the process of removing unwanted hair from a part of the body by applying wax and peeling off the wax and hairs together.

simply  $(144 \text{ power of } 3/2)^{-1/2}$

$2\sqrt{3}$

**Right angle traingle program:**

```
// C++ program to implement

// the above approach

#include <bits/stdc++.h>

using namespace std;

// Function to check if right-angled
// triangle can be formed by the
// given coordinates

void checkRightAngled(int X1, int Y1,
                      int X2, int Y2,
                      int X3, int Y3)

{
    // Calculate the sides

    int A = (int)pow((X2 - X1), 2)
        + (int)pow((Y2 - Y1), 2);

    int B = (int)pow((X3 - X2), 2)
        + (int)pow((Y3 - Y2), 2);

    int C = (int)pow((X3 - X1), 2)
        + (int)pow((Y3 - Y1), 2);

    // Check Pythagoras Formula

    if ((A > 0 and B > 0 and C > 0)
        and (A == (B + C) or B == (A + C)))
```

```

        or C == (A + B)))
        cout << "Yes";
    }

else
    cout << "No";
}

// Driver Code
int main()
{
    int X1 = 0, Y1 = 2;
    int X2 = 0, Y2 = 14;
    int X3 = 9, Y3 = 2;

    checkRightAngled(X1, Y1, X2,
                      Y2, X3, Y3);

    return 0;
}

```

Fibonacci series in c:

```

include<stdio.h>

int main()
{
    int n1=0,n2=1,n3,i,number;

```

```
printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n%d %d",n1,n2);//printing 0 and 1

for(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed

{

n3=n1+n2;

printf(" %d",n3);

n1=n2;

n2=n3;

}

return 0;
```

A binary tree with the property that the values at each node is at least as large as the values at its children is known as

**A**

Binary search tree

**B**

AVL tree

**C**

Completely balanced tree

**D**

Heap

answer:Heap

**Sort array in c:**

```
1. #include <stdio.h>
2. void main()
3. {
4.
5.     int i, j, a, n, number[30];
6.     printf("Enter the value of N \n");
7.     scanf("%d", &n);
8.
9.     printf("Enter the numbers \n");
10.    for (i = 0; i < n; ++i)
11.        scanf("%d", &number[i]);
12.
13.    for (i = 0; i < n; ++i)
14.    {
15.
16.        for (j = i + 1; j < n; ++j)
17.        {
18.
19.            if (number[i] > number[j])
20.            {
```

```
21.
22.         a = number[i];
23.         number[i] = number[j];
24.         number[j] = a;
25.
26.     }
27.
28. }
29.
30. }
31.
32.     printf("The numbers arranged in ascending order are given below \n");
33.     for (i = 0; i < n; ++i)
34.         printf("%d\n", number[i]);
35.
36. }
```

### Count digits in c:

```
#include <stdio.h>

int main() {

    long long n;

    int count = 0;

    printf("Enter an integer: ");

    scanf("%lld", &n);

    // iterate until n becomes 0
```

```

// remove last digit from n in each iteration

// increase count by 1 in each iteration

while (n != 0) {

    n /= 10;    // n = n/10

    ++count;

}

printf("Number of digits: %d", count);

}

```

### **Finding max and min in an array:**

```

#include <stdio.h>

void main()

{
    int arr1[100];

    int i, mx, mn, n;

    printf("\n\nFind maximum and minimum element in an array :\n");

    printf("-----\n");

```

```
printf("Input the number of elements to be stored in the array :");
```

```
scanf("%d",&n);
```

```
printf("Input %d elements in the array :\n",n);
```

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

```
{
```

```
printf("element - %d : ",i);
```

```
scanf("%d",&arr1[i]);
```

```
}
```

```
mx = arr1[0];
```

```
mn = arr1[0];
```

```
for(i=1; i<n; i++)
```

```
{
```

```
if(arr1[i]>mx)
```

```
{
```

```
mx = arr1[i];
```

```
}
```

```

if(arr1[i]<mn)
{
    mn = arr1[i];
}

printf("Maximum element is : %d\n", mx);

printf("Minimum element is : %d\n\n", mn);

}

```

### Median program in c:

```
#include <stdio.h>
```

```

void swap(int *p,int *q) {

    int t;

    t=*p;
    *p=*q;
    *q=t;

}

```

```
void sort(int a[],int n) {  
    int i,j,temp;  
  
    for(i = 0;i < n-1;i++) {  
        for(j = 0;j < n-i-1;j++) {  
            if(a[j] > a[j+1])  
                swap(&a[j],&a[j+1]);  
        }  
    }  
  
    int main() {  
        int a[] = {6,3,8,5,1};  
        int n = 5;  
        int sum,i;  
  
        sort(a,n);  
  
        n = (n+1) / 2 - 1;      // -1 as array indexing in C starts from 0  
  
        printf("Median = %d ", a[n]);  
    }  
}
```

```
    return 0;  
}  
  
}
```

When a function in the base class is redefined in the inherited class is called? **Overriding is the word when a function belonging to the base class is redefined in the inherited class which is part of Java language. It is a language feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super classes or parent classes.**

**Selection sort:**

```
#include <stdio.h>  
  
int main() {  
  
    int arr[10]={6,12,0,18,11,99,55,45,34,2};  
  
    int n=10;  
  
    int i, j, position, swap;  
  
    for (i = 0; i < (n - 1); i++) {  
  
        position = i;  
  
        for (j = i + 1; j < n; j++) {  
  
            if (arr[position] > arr[j])  
  
                position = j;  
  
        }  
  
        if (position != i) {  
  
            swap = arr[i];  
  
            arr[i] = arr[position];  
  
            arr[position] = swap;  
  
        }  
    }  
}
```

```
arr[i] = arr[position];  
  
arr[position] = swap;  
  
}  
  
}  
  
for (i = 0; i < n; i++)  
  
printf("%d\t", arr[i]);  
  
return 0;  
  
}
```

## Question 1

The function/method patternPrint accepts an argument num, an integer. The function/method patternPrint prints num lines in the following pattern.

For example, num=4 the pattern should be

```
1
11
111
1111
```

The function/method patternPrint compiles successfully but fails to print the desired result for some test cases due to incorrect implementation of the function/method. Your task is to fix the code so that it passes all the test cases.

```
void patternPrint(int num)
{
    int print=1,i,j;
    for(i=0;i<num;i++)
    {
        for(j=0;j<=i;j++);
        {
            printf("%d ",print);
        }
        printf("\n");
    }
}
```

## Solution:

```
void patternPrint(int num)
{
    int print=1,i,j;
    for(i=0;i<num;i++)
    {
        for(j=0;j<=i;j++)
        {
            printf("%d",print);
        }
    }
}
```

```
    }  
    printf("\n");  
}  
}
```

## Question 2

Lisa always forgets her birthday which is on the 5th of July. So, develop a function/method which will be helpful to remember her birthday.

The function/method `checkBirthDay` returns an integer '1' if it is her birthday else returns 0. The function/method `checkBirthDay` accepts two arguments - month, a string representing the month of her birthday and day an integer representing the date of her birthday.

The function/method `checkBirthDay` compiles successfully but fails to print the desired result for some test cases due to incorrect implementation of the function/method. Your task is to fix the code so that it passes all the test cases.

```
int checkBirthDay(char* month,int day)
```

```
{  
    if((strcmp(month,"July"))|| (day==5))  
        return 1;  
    else  
        return 0;  
}
```

### Solution:

```
int checkBirthDay(char* month, int day)
{
    if((strcmp(month,"July")==0) && (day==5))
        return 1;
    else
        return 0;
}
```

## Question: 3

The function/method **allExponent** returns a real number representing the result of exponentiation of base raised to power exponent for all input values. The function/method **allExponent** accepts two arguments **baseValue**, an integer representing the base and **exponentValue**, an integer representing the exponent.

The incomplete code in the function/method **allExponent** works only for positive values of the exponent. You must complete the code and make it work for negative values of exponent as well.

Another function/method **positiveExponent** uses an efficient way for exponentiation but accepts only positive exponent values. You are supposed to use this function/method complete the code in **allExponent** function/method.

### Helper Description

The following function is used to represent a positive Exponent and is already implemented the default code (Do not write this definition again in your code):

```
int positiveExponent (int baseValue, int exponentValue)
{
    /*It calculates the Exponent for the positive value of exponentValue
    This can be called as –
    int res (float)positiveExponent (baseValue, exponentValue);* /
}
```

### Solution:

```
Float allExponent(int baseValue, int exponentValue)
{
    float res = 1;
    if(exponentValue>=0){
        res = (float)positiveExponent(baseValue,exponentValue);
    }
    else{
```

```
    res = 1/(float)positiveExponent(baseValue,exponentValue);
}
return res;
}
```

**Testcase 1:**

Input: 5, 2

Expected Return Value: 25.00000

**Testcase 2:**

Input: 5, -2

Expected Return Value: 0.04000

**Question: 4**

You are given a predefined structure Point and also a collection of related functions/methods that can be used to perform some basic operations on the structure.

You must implement the function/method **isTriangle** which accepts three points P1, P2, P3 as inputs and checks whether the given three points form a triangle.

If they form a triangle. the function/method returns an integer 1.

Otherwise, it returns an integer 0. **Helper Description**

The following structure is used to represent point and is already implemented in the default code (Do not write these definitions again in your code):

**Code:**

```
struct point;
```

```
typedef struct point{
```

```
    int X;
```

```
    int Y;
```

```
}Point;
```

```
double Point calculateDistance(Point *point1, Point *point2)
```

```
int isTriangle(Point *p1, Point *p2, Point *p3)
```

```
{
```

```
//write your code here
```

```
}
```

## **Solution:**

```
int isTriangle(Point *p1, Point *p2, Point *p3)
{
    double d1 = calculateDistance(p1,p2);
    double d2 = calculateDistance(p2,p3);
    double d3 = calculateDistance(p3,p1);
    if((d1+d2>d3) && (d1+d3>d2) &&(d2+d3>d1))
        return 1;
    else
        return 0;
}
```

### **Testcase 1:**

Input:

(3,4),  
(2,1),  
(1, 5)

Expected Return Value: 1

### **Testcase 2:**

Input:

(1,-1),  
(0,-1),  
(- 1, - 1)

Expected Return Value: 0

## Question 5

You are required to fix all syntactical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required. **Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method `matrixSum` returns an integer representing the sum of elements of the input matrix. The function/method `matrixSum` accepts three arguments - `rows`, an integer representing the number of rows of the input matrix, `columns`, an integer representing the number of columns of the input matrix and `matrix`, a two-dimensional array representing the input matrix.

The function/method `matrixSum` compiles unsuccessfully due to syntactical error. Your task is to debug the program so that it passes all test cases.

```
1 // You can print the values to stdout for debugging
2 int matrixSum(int rows, int columns, int **matrix)
3 {
4     int i, j, sum=0;
5     for(i=0;i<rows;i++)
6     {
7         for(j=0;j<columns;j++)
8             sum += matrix(i)(j);
9     }
10 }
11 }
```

### Testcase 1:

#### Input:

```
3, 3,
[[3, 2, 1],
 [4, 6, 5],
 [7, 8, 9]]
```

#### Expected Return Value:

45

---

### Testcase 2:

#### Input:

```
3, 3,
[[3, 12, 10],
 [14, 61, 51],
 [27, 84, 95]]
```

#### Expected Return Value:

357

## Solution:

```
int matrixSum(int rows, int columns, int **matrix)
{
    int i, j,sum=0;
    for(i=0; i<rows; i++)
    {
        for(j=0; j<columns; j++)
            sum += matrix[i][j];
    }
    return sum;
}
```

## Question 6

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **selectionSortArray** performs an in-place selection sort on the given input list which will be sorted in ascending order.

The function/method **selectionSortArray** accepts two arguments - *len*, an integer representing the length of the input list and *arr*, a list of integers representing the input list, respectively.

The function/method **selectionSortArray** compiles successfully but fails to get the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

**Note:**

In this particular implementation of selection sort, the smallest element in the list is swapped with the element at first index, the next smallest element is swapped with the element at the next index and so on.

```
1 // You can print the values to stdout for debugging
2 void selectionSortArray(int len, int* arr)
3 {
4     int x=0, y =0;
5     for(x=0; x<len; x++){
6         int index_of_min = x;
7         for(y=x; y<len; y++){
8             if(arr[index_of_min]>arr[y]){
9                 index_of_min = y;
10            }
11        }
12        int temp = arr[x];
13        arr[x] = arr[index_of_min];
14        arr[index_of_min] = temp;
15    }
16 }
17 }
```

**Testcase 1:**

**Input:**

10, [3, 6, 4, 1, 7, 9, 1, 3, 12, 15]

**Expected Return Value:**

1 1 3 3 4 6 7 9 12 15

**Testcase 2:**

**Input:**

9, [3, 3, 3, 3, 3, 3, 3, 3, 3]

**Expected Return Value:**

3 3 3 3 3 3 3 3 3

## Solution:

```
void selectionSortArray(int len, int *arr)
{
    int x=0, y=0;
    for(x=0; x<len; x++)
    {
        int index_of_min = x;
        for(y=x; y<len; y++)
        {
            if(arr[index_of_min]>arr[y])
            {
                index_of_min=y;
            }
        }
    }
}
```

```

        If(index_of_min!=x)
        {
            int temp = arr[x];
            arr[x] = arr[index_of_min];
            arr[index_of_min] = temp;
        }
    }
}

```

### Question 7

The function/method **descending SortArray** performs an in-place sort on the given input list which will be sorted in descending order.

The function/method **descending SortArray** accepts two arguments - **len**, an integer representing **the length of the input list and arr, a list of integers representing the input list, respectively**.

The function/method **descending SortArray** compiles successfully but fails to get the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

#### ***Given function:-***

```
void descendingSortArray(int len,int *arr)
```

```

{
    int small,pos,i,j,temp;
    for(i=0;i<=len-1;i++)
    {
        for(j=i;j<len;j++)
        {
            temp=0;
```

```
if(arr[i]>arr[j])  
{  
    temp=arr[i];  
    arr[i]=arr[j];  
    arr[j]=temp;  
}  
}  
}
```

***Solution:-***

```
void descendingSortArray(int len,int *arr)  
{  
    Int i,j,temp;  
    for(i=0;i<=len-1;i++)  
    {  
        for(j=i;j<len;j++)  
        {  
            if(arr[i]<arr[j])  
            {  
            }
```

```
    temp=arr[i];  
  
    arr[i]=arr[j];  
  
    arr[j]=temp;  
  
}  
  
}  
  
}
```

## Question 8

The function/method **sameElementCount** returns an integer representing the number of elements of the input list which are even numbers and equal to the element to its right. For example, if the input list is [4 4 418 41122] then the function/method should return the output 3 as it has three similar groups ie, (4, 4). (4, 4), (2, 2).

The function method **sameElementCount** accepts two arguments - size, an integer representing the size of the input list and input List, a list of integers representing the input list.

The function/method compiles successfully but fails to return the desired result for some test cases due to incorrect implementation of the function/method **sameElementCount** Your task is to fix the code so that it passes all the test cases.

### Note:

In a list, an element at index  $i$  is considered to be on the left of index  $i+1$  and to the right of index  $i-1$ . The last element of the input list does not have any element next to it which makes it incapable to satisfy the second condition and hence should not be counted.

**Solution:-**

```
int sameElementCount(int size,int *inputList)
{
    for(i=0;i<=size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
        {
            count++;
        }
    }
    return count;
}
```

**Test cases 1:-**

**Input:-**

11,  
[1,5,5,2,2,7,8,6,6,9,10]

**Expected return value:-**

2

## Test case:-2

### Input:

5,

[13,12,12,13,14]

### Expected return value:

1

## Question 9

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **countOccurrence** return an integer representing the count of occurrences of given value in the input list.

The function/method **countOccurrence** accepts three arguments - *len*, an integer representing the size of the input list, *value*, an integer representing the given value and *arr*, a list of integers, representing the input list.

The function/method **countOccurrence** compiles successfully but fails to return the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 int countOccurrence( int len, int value, int *arr)
3 {
4     int i=0, count = 0;
5     while(i<len){
6         if(arr[i]==value)
7             count += 1;
8     }
9     return count;
10 }
11 }
```

**Testcase 1:**  
**Input:**  
7, 3, [2, 3, 4, 3, 5, 6, 7]

**Expected Return Value:**  
2

**Testcase 2:**  
**Input:**  
1, 2, [9]  
**Expected Return Value:**  
0

```
1 // You can print the values to stdout for debugging
2 int countOccurrence( int len, int value, int *arr)
3 {
4     int i=0, count = 0;
5     while(i<len){
6         if(arr[i]==value)
7             count += 1;
8     }
9     return count;
10 }
11 }
```

Activate Window  
Go to Settings to ac

## Solution:

```

int countOccurrence(int len, int value,int *arr)
{
int i=0,count=0;
while(i<len)
{
if(arr[i]==value)
{
count+=1;
}
i++;
}
return count;
}

```

## Question 10

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to complete the code as is given implementation. We **do not** expect you to modify the approach. The function/method *median* accepts two arguments - *size* and *inputList*, an integer representing the length of a list and a list of integers, respectively.

The function/method *median* is supposed to calculate and return an integer representing the median of elements in the input list. However, the function/method *median* works only for odd-length lists because of incomplete code.

You must complete the code to make it work for even-length lists as well. A couple of other functions/methods are available, which you are supposed to use inside the function/method *median* to complete the code.

### Helper Description

The following function is used to represent a *quick\_select* and is already implemented in the default code (Do not write this definition again in your code):

```

int quick_select(int* inputList, int start_index, int end_index, int median_order)
{
    /*It calculate the median value
    This can be called as -
    quick_select(inputList, start_index, end_index, median_order)
    where median_order is the half length of the inputList
}

```

### Testcase 1:

**Input:**  
5,  
[2, 40, 23, 52, 37]

**Expected Return Value:**  
37.00

### Testcase 2:

**Input:**  
6,  
[-24, -16, -8, -4, -54, -1]

**Expected Return Value:**  
-12.00

```

1 // You can print the values to stdout for debugging
2 float median(int size, int * inputList)
3 {
4     int start_index = 0;
5     int end_index = size-1;
6     float res = -1;
7     if(size%2==0) // odd size inputList
8     {
9         int median_order = ((size+1)/2);
10        res = (float)quick_select(inputList, start_index, end_index);
11    }
12    else // even size inputList
13    {
14        // Write code here
15    }
16    return res;
17 }
18
19

```

```

1 // You can print the values to stdout for debugging
2 float median(int size, int * inputList)
3 {
4     int start_index = 0;
5     int end_index = size-1;
6     float res = -1;
7     if(size%2==0) // odd size inputList
8     {
9         int median_order = ((size+1)/2);
10        res = (float)quick_select(inputList, start_index, end_index);
11    }
12    else // even size inputList
13    {
14        // Write code here
15    }
16    return res;
17 }
18
19

```

Acti  
Go to

## Solution

```
float median(int size,int *inputList)
{
int start_index=0;
int end_index=size-1;
float res=-1;
if(size%2!=0)
{
int med_1=size/2;
int med_2=(size/2)+1;
res=((float)quick_select(inputList,start_index,end_index,med_1));
}
else
{
//enter your code here
res=((float)quick_select(inputList,start_index,end_index,med_1)+(float)
quick_select(inputList,st
art_index,end_index,med_2))/2;
}
return res;
}
```

## Question 11

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporate any additional library methods.

The function/method **manchester** print space-separated integers with the following property: for each element in the input list *arr*, if the bit *arr[i]* is the same as *arr[i-1]*, then the element of the output list is 0. If they are different, then its 1. For the first bit in the input list, assume its previous bit to be 0. This encoding is stored in a new list.

The function/method **manchester** accepts two arguments - *len*, an integer representing the length of the list and *arr* and *arr*, a list of integers, respectively. Each element of *arr* represents a bit - 0 or 1

For example - if *arr* is {01001110}, the function/method should print an list {01101001}.

The function/method compiles successfully but fails to print the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void manchester(int len, int* arr)
3 {
4     int* res = (int*)malloc(sizeof(int)*len);
5     res[0] = arr[0];
6     for(int i = 1; i < len; i++){
7         res[i] = (arr[i]==arr[i-1]);
8     }
9     for(int i = 0; i < len; i++)
10        printf("%d ",res[i]);
11 }
```

#### Testcase 1:

##### Input:

6, [1, 1, 0, 0, 1, 0]

##### Expected Return Value:

1 0 1 0 1 1

---

#### Testcase 2:

##### Input:

8, [0, 0, 0, 1, 0, 1, 1, 1]

##### Expected Return Value:

0 0 0 1 1 1 0 0

## Solution:

```
void manchester(int len,int *arr)
{
    int* res=(int)malloc(sizeof(int)*len);
    res[0]=arr[0];
    for(int i=1;i<len;i++)

```

```

{
    res[i]=!(arr[i]==arr[i-1]);
}
for(int i=0;i<len;i++)
    printf("%d ",res[i]);
}

```

## Question 12

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We **do not** expect you to modify the approach.

You are given a predefined structure/class **Point** and also a collection of related functions/methods that can be used to perform some basic operations on the structure.

The function/method **isRightTriangle** returns an integer '1', if the points make a right-angled triangle otherwise return '0'.

The function/method **isRightTriangle** accepts three points - *P1*, *P2*, *P3* representing the input points.

You are supposed to use the given function to complete the code of the function/method **isRightTriangle** so that it passes all test cases.

### Helper Description

The following structure is used to represent point and is already implemented in the default code (Do not write these definitions again in your code):

```

struct point;
typedef struct point
{
    int X;
    int Y;
}Point;
double Point_calculateDistance(Point *point1, Point *point2)
{

```

```

1 // You can print the values to stdout for debugging
2 int isRightTriangle(Point *P1, Point *P2, Point *P3)
3 {
4     // write your code here
5 }
6
7

```

**Testcase 1:****Input:**

```
(2, -2),  
(-2, -2),  
(1, -2)
```

**Expected Return Value:**

```
0
```

---

**Testcase 2:****Input:**

```
(-1, 0),  
(2, 0),  
(-1, -4)
```

**Expected Return Value:**

```
1
```

**Solution:**

```
int isRightTriangle(Point *P1,Point *P2,Point *P3)  
{  
    double d1,d2,d3;  
    d1=Point_calculateDistance(P1,P2);  
    d2=Point_calculateDistance(P2,P3);  
    d3=Point_calculateDistance(P3,P1);  
    if((pow(d1,2)+pow(d2,2)==pow(d3,2))||pow(d2,2)+pow(d3,2)==pow(d1,2)||pow((d3,2)+pow(d1,2)==pow(d2,2))  
    {  
        return 1;  
    }  
    else  
        return 0;  
}
```

**Question 13**

You are required to fix all syntactical errors in the given code. You can click on **Compile & Run** anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be

logically/syntactically correct and pass all testcases. Do not write the main function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporate any additional library methods.

The function/method multiplyNumber returns an integer representing the multiplicative product of the maximum two of three

input numbers. The function/method multiplyNumber accepts three integers- numA, numB and numc, representing the input numbers.

The function/method multiplyNumber compiles unsuccessfully due to syntactical error. Your task is to debug the code so that it passes all the test cases.

### **Solution:**

```
void drawPrintPattern(int num)
{
    int i,j,print=1;
    for (i=1;i<=num;i++)
    {
        for (j=1;j<=2*i;j++)
        {
            printf("%d ",print);
        }
        printf ("\n");
    }
}
```

**Testcase 1:**

Input: 5, 7, 4

Expected Return Value:

35

**Testcase 2:**

Input: 11, 12, 13

Expected Return Value:

156

**Question 14**

You are required to fix all logical errors in the given code. You can click on Compile & Run anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation.

We do not expect you to modify the approach or incorporate any additional library methods. The function/method `drawPrintPattern` accepts `num`, an integer. The function/method `drawPrintPattern` prints the first `num` lines of the pattern shown below.

For example, if `num = 3`, the pattern should be:

11

1111

111111

The function/method drawPrintPattern compiles successfully but fails to get the desired result for some test cases due to incorrect implementation of the function/method. Your task is to fix the code so that it passes all the test cases.

**Solution:**

```
int multiplyNumber(int numA,int numB,int numC)
{
    int result,min,max,mid;
    max=(numA>numB)?((numA>numC)?numA:numC):((numB>numC)?numB:numC);
    min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
    mid=(numA+numB+numC)-(min+max);
    result=(max*mid);
    return result;
}
```

**Testcase 1:**

Input: 4

Expected Return Value:

```
11
1111
111111
1111111
```

**Testcase 2:**

Input: 1

Expected Return Value:

```
11
```

**Question 15**

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main* function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We **do not** expect you to modify the approach.

The function/method ***difference\_in\_dates*** return an integer representing the difference between given two dates. The difference shall always be either a positive number or zero.

The function/method ***difference\_in\_dates*** accepts two argument - *date1* and *date2*, representing given two Dates.

Developers at ABC Technologies already use a predefined structure ***Date*** containing day, month, and year as members. A collection of functions/methods for performing some common operations on dates is also available. You are supposed to make use of these functions/methods to calculate and return the difference.

```
1 // You can print the values to stdout for debugging
2 int difference_in_dates(Date *date1, Date *date2)
3 {
4     // write your code here
5 }
6
```

## **Solution:**

```
int difference_in_Dates(Date *Date1, Date *Date2)
```

```
{
```

```
    Return *Date2-*Date1;
```

```
}
```

## **Question 16**

You are required to fix all syntactical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **replaceMinMax** is supposed to replace all the even elements of the input list with the maximum element of the list, also replace all the odd elements of arr with the minimum element of the list.

The function/method **replaceMinMax** accepts two arguments - *size*, an integer representing the size of the input list and *arr*, a list of integers representing the input list.

The function/method **replaceMinMax** compiles unsuccessfully due to syntactical error. Your task is to debug the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void replaceMinMax(int size, int* arr)
3 {
4     int i;
5     if(size>0)
6     {
7         int max = arr[0];
8         int min = arr[0];
9         for(i=0;i<size;i++)
10        {
11            if(max<arr[i])
12            {
13                max = arr[i];
14            }
15            else if(min > arr[i])
16            {
17                min = arr[i];
18            }
19        }
20        for(i=0;i<size;i++)
21        {
22            if(arr[i] % 2 == 0)
23                arr[i]=max;
24            else
25                arr[i]=min;
26        }
27    }
28 }
```

## Solution:

Void replaceMinMax(int size, int\* arr)

{

    int i;

    if(size>0)

    {

        int max = arr[0];

        int min = arr[0];

        for(i=0; i<size;i++)

        {

            if(max<arr[i])

```
{  
    max = arr[i];  
}  
else  
{  
    if(min>arr[i])  
    {  
        min= arr[i]  
    }  
}  
for(i=0;i<size;i++)  
{  
    if(arr[i] %2 ==0)  
    {  
        arr[i]=max;  
    }  
    else  
{  
        arr[i] = min;  
    }  
}
```

}

}

## Question 17

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **printPattern** accepts an argument *num*, an integer. The function/method **printPattern** print the first *num* lines of the pattern as shown below.

For example, if *num* = 3, the pattern should be:

```
11
2222
333333
```

The function/method **printPattern** compiles successfully but fails to print the desired result for some test cases. Your task is to debug the code so that it passes all the test cases.

**Testcase 1:**

**Input:**

4

**Expected Return Value:**

```
1 1
2 2 2
3 3 3 3 3
4 4 4 4 4 4 4
```

**Testcase 2:**

**Input:**

5

**Expected Return Value:**

```
1 1
2 2 2
3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5 5
```

```
1 #include<stdio.h>
2 void printPattern(int num)
3 {
4     int i,j;
5     for(i=1;i<=num;i++)
6     {
7         for(j=1;j<=2*i;j++)
8         {
9             printf("%d ",j);
10        }
11        printf("\n");
12    }
13 }
14 }
```

```
1 #include<stdio.h>
2 void printPattern(int num)
3 {
4     int i,j;
5     for(i=1;i<=num;i++)
6     {
7         for(j=1;j<=2*i;j++)
8         {
9             printf("%d ",j);
10        }
11        printf("\n");
12    }
13 }
14 }
```

QUEST

## Solution:

```
#include <stdio.h>
void printPattern(int num)
{
    int i,j;
    for(i=1;i<=num;i++)
    {
        for(j=1;j<=2*i;j++)
        {
            printf("%d ",i);
        }
    }
}
```

```

        printf("\n");
    }
}

```

## Question 18

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method *printFibonacci* accepts an integer *num*, representing a number.

The function/method *printFibonacci* prints first *num* numbers of the Fibonacci series.

For example, given input 5, the function should print the string "0 1 1 2 3" (without quotes).

The function/method compiles successfully but fails to give the desired result for some test cases. Your task is to debug the code so that it passes all the test cases.

**Testcase 1:**

**Input:**  
23

**Expected Return Value:**

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1

**Testcase 2:**

**Input:**  
14

**Expected Return Value:**

0 1 1 2 3 5 8 13 21 34 55 89 144 233

```

1 void printFibonacci(int num)
2 {
3     int i;
4     long sum=0;
5     long num1 = 0;
6     long num2 = 1;
7     for ( i = 1; i < num; ++i)
8     {
9         printf("%ld ", num1);
10        sum = num1 + num2;
11        num2 = sum;
12        num1 = num2;
13    }
14 }

```

```

1 void printFibonacci(int num)
2 {
3     int i;
4     long sum=0;
5     long num1 = 0;
6     long num2 = 1;
7     for ( i = 1; i < num; ++i)
8     {
9         printf("%ld ", num1);
10        sum = num1 + num2;
11        num2 = sum;
12        num1 = num2;
13    }
14 }

```

## Solution:

```

#include <stdio.h>
void printFibonacci (int num)
{
    int i;
    long sum = 0;
    long num1 = 0;
    long num2= 1;
    for (i = 1; i <= num; ++i)
    {
        printf("%ld ", num1);
        sum = num1 + num2;
        num1 = num2;
        num2 = sum;
    }
}

```

```

    num1 = num2;
    num2 = sum;
}
}

```

### Question 19

The function/method manchester print space seperated integers with the following property.

For each element in the input arr,a counter is incremented if the bit arr[i] is same as arr[i-1]. Then the increment counter value is added to the output array to store the result.

If the bit arr[i] and arr[i-1] are not same then the output is added with 0. For the first bit in the arr, assume its previous bit to be 0.

For example, if arr is {0,1,0,0,1,1,1,0} then the output would be 1 0 0 2 0 3 4 0.

#### Solution:

```

#include<stdbool.h>
void manchester(int size, int* arr){
    bool result;
    int* res=(int*)malloc(sizeof(int)*size);
    int count=0;
    for(int i=0;i<size;i++)
    {
        if(i==0)
            result= (arr[i]==0);
        else
            result= (arr[i]==arr[i-1]);
        res[i]=(result)?(++count):0;
    }
    for(int i=0;i<size;i++)
    {
        printf("%d ",res[i]);
    }
}

```

**Test Case 1:**

Input:

6

[1,1,0,0,1,0]

Output:

0 1 0 2 0 0

**Test Case 2:**

Input:

8

[0,0,0,1,0,1,1,1]

Output:

1 2 3 0 0 0 4 5

**Question 20**

The function/method reverseHalfArray modify the inputList by reversing the input list from the second half. The function takes size and array as arguments.

**Solution:**

```
void reverseHalfArray(int size, int* inputList){  
    int i,temp;  
    for(i=(size/2);i<size;i++)  
    {  
        temp=inputList[size-1];  
        inputList[size-1]=inputList[i];  
        inputList[i]=temp;  
        size-=1;  
    }  
}
```

**Test Case 1:**

Input:

7

[1,2,3,4,5,6,7]

Output:

1 2 3 7 6 5 4

## Test Case2:

Input:

4

[2,8,4,6]

Output:

2 8 6 4

## Question 21

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We **do not** expect you to modify the approach. You are given a predefined structure/class `Point` and also a collection of related functions/methods that can be used to perform some basic operations on the structure.

The function/method `isRightTriangle` returns an integer '1', if the points make a right-angled triangle otherwise return '0'.

The function/method `isRightTriangle` accepts three points - `P1`, `P2`, `P3` representing the input points.

You are supposed to use the given function to complete the code of the function/method `isRightTriangle` so that it passes all test cases.

### Helper Description

The following structure is used to represent point and is already implemented in the default code (Do not write these definitions again in your code):

```
1 // You can print the values to stdout for debugging
2 int isRightTriangle(Point *P1, Point *P2, Point *P3)
3 {
4     // write your code here
5 }
6
7 }
```

## Solution:

```
int isRightTriangle(Point *P1, Point *P2, Point *P3)
{
    double d1, d2, d3;
    d1 = Point_calculateDistance(P1, P2);
    d2 = Point_calculateDistance(P2, P3);
    d3 = Point_calculateDistance(P3, P1);
    if((pow(d1, 2) + pow(d2, 2) == pow(d3, 2)) || (pow(d2, 2) + pow(d3, 2) == pow(d1, 2)) || (pow(d3, 2) + pow(d1, 2) == pow(d2, 2)))
    {
```

```

        return 1;
    }
    else return 0;
}

```

## Question 22

Problem | Test Cases | Output |

Compile and Run



You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method `printCharacterPattern` accepts an integer `num`. It is supposed to print the first `num` ( $0 \leq num \leq 26$ ) lines of the pattern as shown below.

For example, if `num = 4`, the pattern is:

```

a
ab
abc
abcd

```

The function/method compiles successfully but fails to print the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```

1 // You can print the values to stdout for debugging
2 void printCharacterPattern(int num){
3     int i, j;
4     char ch='a';
5     char print;
6     for(i=0;i<num;i++){
7         print = ch;
8         for(j=0;j<=i;j++)
9             printf("%c",ch++);
10            printf("\n");
11    }
12 }

```

## Solution:

```

void printCharacterPattern(int num)
{
    int i,j;
    char ch='a';
    char print;
    for(i=0;i<=i;j++)
    {
        printf("%c",print++);
    }
}

```

```
    printf("\n");
}
}
```

### Question 23

You are required to fix all logical errors in the given code. You can click on Compile & Run anytime to check the compilation/execution status of the program. You can use printf() to debug your code. The submitted code should be logically/syntactically correct and pass all testcases

**Code Approach:** For this question, you will need to correct the given implementation. We do not expect you to modify the approach.or incorporate any additional library methods.

The function/method countDigits return an integer representing the remainder when the given number is divided by the number of digits in it. The function/method countDigits accepts an argument - num, an integer representing the given number.

The function/method countDigits compiles successfully but fails to print the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases

**Solution:**

```
int countDigits(int num)
```

```
{
```

```
    int count=0;
```

```
    int k=num;
```

```
    while(num!=0)
```

```
{
```

```
        num=num/10;
```

```
        ++count;
```

```
    }  
    return (k%count);  
}
```

### Testcase 1:

Input:

782

Expected Return Value: 2

### Testcase 2:

Input:

21340

Expected Return Value: 0

## Question 24

You are required to complete the given code. You can click on Compile & Run anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the main function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We do not expect you to modify the approach.

The function/method difference in dates return an integer representing the difference between given two dates. The difference shall always be either a positive number or zero

The function/method difference in dates accepts two argument-datel and date2, representing given two Dates

Developers at ABC Technologies already use a predefined structure Date containing day, month, and year as members. A collection of functions/methods for performing some common operations on dates is also available. You are supposed to make use of these functions/methods to calculate and return the difference.

**Solution:**

```
int difference_in_dates (Date *date1, Date *date2)
{
    long int n1 = date1.y* 365+ date1.d;
    for (int i = 0; i < date1.m - 1; i++)
        n1 += monthDays[i];
    n1 += countLeapYears(date1);
    long int n2 = date2.y* 365+ date2.d;
    for (int i = 0; i < date2.m-1; i++)
        n2 += monthDays[i];
    n2 += countLeapYears(date2);
    return (n2 - n1);
}
```

**Testcase 1:**

Input:

[day:82, month:05, year:2013),

[day:82, month:06, year:2013)

Expected Return Value:

31

**Testcase 2:**

Input:

[day:01, month:06, year: 2011), (day:01, month:06, year:2012)

Expected Return Value:

366

## Question 25

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **arrayReverse** modify the input list by reversing its element. The function/method **arrayReverse** accepts two arguments - *len*, an integer representing the length of the list and *arr*, list of integers representing the input list, respectively.

For example, if the input list *arr* is {20 30 10 40 50}, the function/method is supposed to print {50 40 10 30 20}.

The function/method **arrayReverse** compiles successfully but fails to get the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void arrayReverse(int len, int* arr)
3 {
4     int i, temp, originalLen=len;
5     for(i=0;i<originalLen/2;i++){
6         temp = arr[len-1];
7         arr[len-1] = arr[i];
8         arr[i] = temp;
9         len -= 1;
10    }
11 }
```

**Testcase 1:**  
**Input:**

4, [4, 2, 8, 6]

**Expected Return Value:**  
6 8 2 4

**Testcase 2:**  
**Input:**

3, [11, 20, 17]

**Expected Return Value:**  
17 20 11

```
1 // You can print the values to stdout for debugging
2 void arrayReverse(int len, int* arr)
3 {
4     int i, temp, originalLen=len;
5     for(i=0;i<originalLen/2;i++){
6         temp = arr[len-1];
7         arr[len-1] = arr[i];
8         arr[i] = temp;
9         len -= 1;
10    }
11 }
```

## Solution:

```
void arrayReverse(int len,int *arr)
{
int i,temp,originalLen=len;
```

```

for(i=0;i<=originalLen/2;i++)
{
temp=arr[len-1];
arr[len-1]=arr[i];
arr[i]=temp;
len-=1;
}
}

```

## Question 26

You are required to fix all syntactical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *print* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main*/function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **countElement** return an integer representing the number of elements in the input list which are greater than twice the input number K. The function/method **countElement** accepts three arguments - *size*, an integer representing the size of the input list, *numK*, an integer representing the input number K and *inputList*, a list of integers representing the input list, respectively.

The function/method **countElement** compiles unsuccessfully due to syntactical error. Your task is to fix the code so that it passes all the test cases.

```

1 // You can print the values to stdout for debugging
2 int countElement(int size, int numK, int *inputList)
3 {
4     int i, count=0;
5     for(i=0,i<size,i++)
6     {
7         if(inputList[i]>2*numK)
8             count+=1;
9     }
10    return count;
11 }
12
13
14

```

**Testcase 1:**  
**Input:**  
`7, 3,  
[-2, -4, -3, -5, -6, -7, -8]`

**Expected Return Value:**  
`0`

**Testcase 2:**  
**Input:**  
`6, 13,  
[22, 55, 66, 33, 44, 77]`

```

1 // You can print the values to stdout for debugging
2 int countElement(int size, int numK, int *inputList)
3 {
4     int i, count=0;
5     for(i=0,i<size,i++)
6     {
7         if(inputList[i]>2*numK)
8             count+=1;
9     }
10    return count;
11 }
12
13
14

```

## Solution:

```

int countElement(int size,int num,int *inputList)
{
int i,count=0;
for(i=0;i<size;i++)
{
if(inputList[i]>2*numK)
{

```

```

count+=1;
}
}
return count;
}

```

## Question 27

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main* function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **removeElement** prints space separated integers that remains after removing the integer at the given index from the input list.

The function/method **removeElement** accepts three arguments - *size*, an integer representing the size of the input list; *indexValue*, an integer representing given index and *inputList*, a list of integers representing the input list.

The function/method **removeElement** compiles successfully but fails to print the desired result for some test cases due to incorrect implementation of the function/method **removeElement**. Your task is to fix the code so that it passes all the test cases.

**Note:**

**Testcase 1:**  
**Input:**  
9 3  
1 2 3 4 5 6 7 8 9

**Expected Return Value:**  
1 2 3 5 6 7 8 9

**Testcase 2:**  
**Input:**  
6 6  
11 23 12 34 54 32

**Expected Return Value:**  
11 23 12 34 54 32

```

1 // You can print the values to stdout for debugging
2 void removeElement(int size, int indexValue, int *inputList)
3 {
4     int i,j;
5     if(indexValue<size)
6     {
7         for(i=indexValue;i<size-1;i++)
8         {
9             inputList[i]=inputList[i+1];
10        }
11        for(i=0;i<size-1;i++)
12        {
13            printf("%d ",inputList[i]);
14        }
15    }
16    for(i=0;i<size;i++)
17    {
18        printf("%d ",inputList[i]);
19    }

```

```

1 // You can print the values to stdout for debugging
2 void removeElement(int size, int indexValue, int *inputList)
3 {
4     int i,j;
5     if(indexValue<size)
6     {
7         for(i=indexValue;i<size-1;i++)
8         {
9             inputList[i]=inputList[i+1];
10        }
11        for(i=0;i<size-1;i++)
12        {
13            printf("%d ",inputList[i]);
14        }
15    }
16    for(i=0;i<size;i++)
17    {
18        printf("%d ",inputList[i]);
19    }

```

## Solution:

```

void removeElement(int size, int indexValue, int *inputList)
{
    int i,j;

```

```

if(indexValue<size)
{
    for(i=indexValue;i<=size-1;i++)
    {
        inputList[i]=inputList[i+1];
    }
    for(i=0;i<size-1;i++)
        printf("%d ",inputList[i]);
}
else
{
    for(i=0;i<size;i++)
        printf("%d ",inputList[i]);
}
}

```

## Question 28

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as is given implementation. We **do not** expect you to modify the approach.

The function/method `findMaxElement` return an integer representing the largest element in the given two input lists.

The function/method `findMaxElement` accepts four arguments - `len1`, an integer representing the length of the first list, `arr1`, a list of integers representing the first input list, `len2`, an integer representing the length of the second input list and `arr2`, a list of integers representing the second input list, respectively.

Another function/method `sortArray` accepts two arguments - `len`, an integer representing the length of the list and `arr`, a list of integers, respectively and return a list sorted ascending order.

Your task is to use the function/method `sortArray` to complete the code in `findMaxElement` so that it passes all the test cases.

```

1 // You can print the values to stdout for debugging
2 int* sortArray(int len, int* arr)
3 {
4     int i=0,j=0,temp=0;
5     for(i=0;i<len;i++)
6     {
7         for(j=i+1;j<len;j++)
8         {
9             if(arr[i]>arr[j])
10             {
11                 temp = arr[i];
12                 arr[i] = arr[j];
13                 arr[j] = temp;
14             }
15         }
16     }
17     return arr;
18 }
19
20 int findMaxElement(int len1, int* arr1, int len2, int* arr2)
21 {
22     // write your code here
23 }
24

```

```
Testcase 1:  
Input:  
12, [2, 5, 1, 3, 9, 8, 4, 6, 5, 2, 3, 11],  
13, [11, 13, 2, 4, 15, 17, 67, 44, 2, 100, 23]
```

```
Expected Return Value:  
100
```

---

```
Testcase 2:  
Input:  
7, [100, 22, 43, 912, 56, 89, 85]  
8, [234, 123, 456, 234, 890, 181]
```

```
Expected Return Value:  
912
```

```
1 // You can print the values to stdout for debugging  
2 int* sortArray(int len, int* arr)  
3 {  
4     int i=0,j=0,temp=0;  
5     for(i=0;i<len;i++)  
6     {  
7         for(j=i+1;j<len;j++)  
8         {  
9             if(arr[i]>arr[j])  
10            {  
11                temp = arr[i];  
12                arr[i] = arr[j];  
13                arr[j] = temp;  
14            }  
15        }  
16    }  
17    return arr;  
18 }  
19  
20 int findMaxElement(int len1, int* arr1, int len2, int* arr2)  
21 {  
22     // write your code here.  
23 }  
24 }
```

## Solution

```
int* sortArray(int len,int* arr)
```

```
{  
    int i=0,j=0,temp=0;  
    for(i=0;i<len;i++)  
    {  
        for(j=i+1;j<len;j++)  
        {  
            if(arr[i]>arr[j])  
            {  
                temp=arr[i];  
                arr[i]=arr[j];  
                arr[j]=temp;  
            }  
        }  
    }  
}
```

```
return arr;
```

```
}
```

```
Int findMaxElement(int len1, int* arr1, int len2, int* arr2)
```

```
{  
    sortArray(len1,arr1);  
    sortArray(len2,arr2);  
    if(arr1[len1-1]>=arr2[len2-1])  
        return arr1[len1-1];  
    else  
        return arr2[len2-1];
```

}

## Question 29

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method `manchester` print space-separated integers with the following property: for each element in the input list `arr`, if the bit `arr[i]` is the same as `arr[i-1]`, then the element of the output list is 0. If they are different, then its 1. For the first bit in the input list, assume its previous bit to be 0. This encoding is stored in a new list.

The function/method `manchester` accepts two arguments - `len`, an integer representing the length of the list and `arr` and `arr`, a list of integers, respectively. Each element of `arr` represents a bit - 0 or 1.

For example - if `arr` is {0 1 0 0 1 1 0}, the function/method should print an list {0 1 1 0 1 0 0 1}.

The function/method compiles successfully but fails to print the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void manchester(int len, int* arr)
3 {
4     int* res = (int*)malloc(sizeof(int)*len);
5     res[0] = arr[0];
6     for(int i = 1; i < len; i++){
7         res[i] = (arr[i]==arr[i-1]);
8     }
9     for(int i = 0; i<len; i++)
10        printf("%d ",res[i]);
11 }
```

## Solution: -

```
void manchester(int len, int* arr)
{
    int* res = (int*)malloc(sizeof(int)*len);
    res[0] = arr[0];
    for(int i = 1; i < len; i++)
    {
        res[i] = (arr[i]!=arr[i-1]);
    }
    for(int i=0; i<len; i++)
        printf("%d ",res[i]);
}
```

## Question 30

You are required to fix all syntactical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method ***multiplyNumber*** returns an integer representing the multiplicative product of the maximum two of three input numbers. The function/method ***multiplyNumber*** accepts three integers- *numA*, *numB* and *numC*, representing the input numbers.

The function/method ***multiplyNumber*** compiles unsuccessfully due to syntactical error. Your task is to debug the code so that it passes all the test cases.

```

1 // You can print the values to stdout for debugging
2 int multiplyNumber(int numA, int numB, int numC)
3 {
4     int result,min,max,mid;
5     max=(numA>numB)?(numA>numC)?numA:numC:((numB>numC)?numB:numC);
6     min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
7     mid=(numA+numB+numC)-(min+max);
8     result=(max*int mid);
9     return result;
10 }

```

## Solution: -

```

int multiplyNumber(int numA, int numB, int numC)
{
    int result,min,max,mid;
    max=(numA>numB)?((numA>numC)?numA:numC):((numB>numC)?numB:numC);
    min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
    mid=(numA+numB+numC)-(min+max);
    result=(max*mid);
    return result;
}

```

## Question 31

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We **do not** expect you to modify the approach.

The function/method ***median*** accepts two arguments - *size* and *inputList*, an integer representing the length of a list and a list of integers, respectively.

The function/method ***median*** is supposed to calculate and return an integer representing the median of elements in the input list. However, the function/method ***median*** works only for odd-length lists because of incomplete code.

You must complete the code to make it work for even-length lists as well. A couple of other functions/methods are available, which you are supposed to use inside the function/method ***median*** to complete the code.

### Helper Description

The following function is used to represent a quick\_select and is already implemented in the default code (Do not write this definition again in your code):

```

int quick_select(int* inputList, int start_index, int end_index, int median_order)
{
    /*It calculate the median value
    This can be called as -
    quick_select(inputList, start_index, end_index, median_order)
    where median_order is the half length of the inputList
}

```

```

1 // You can print the values to stdout for debugging
2 float median(int size, int * inputList)
3 {
4     int start_index = 0;
5     int end_index = size-1;
6     float res = -1;
7     if(size%2!=0) // odd size inputList
8     {
9         int median_order = ((size+1)/2);
10        res = (float)quick_select(inputList, start_index, end_index);
11    }
12    else // even size inputList
13    {
14        // Write code here
15    }
16    return res;
17 }
18
19

```

Solution:

```
float median(int size,int *inputList)
{
int start_index=0;
int end_index=size-1;
float res=-1;
if(size%2!=0)
{
int median_order=((size+1)/2);
res=(float)quick_select(inputList,start_index,end_index,median_order);
}
else
{
int median_order=((size+1)/2);
res=((float)(quick_select(inputList,start_index,end_index,median_order)+quick_select(inputList,start_index,end_index,median_order))/2;
}
return res;
}
```

## Question 32

You are required to complete the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as is given implementation. We **do not** expect you to modify the approach.

You are given a predefined structure/class `Point` and also a collection of related functions/methods that can be used to perform some basic operations on the structure.

The function/method `isRightTriangle` returns an integer '1', if the points make a right-angled triangle otherwise return '0'.

The function/method `isRightTriangle` accepts three points - `P1, P2, P3` representing the input points.

You are supposed to use the given function to complete the code of the function/method `isRightTriangle` so that it passes all test cases.

**Helper Description**

The following structure is used to represent point and is already implemented in the default code (Do not write these definitions again in your code):

```
struct point;
typedef struct point
{
    int X;
    int Y;
}Point;
double Point_calculateDistance(Point *point1, Point *point2)
```

Solution:

```
int isRightTriangle(Point *P1,Point *P2,Point *P3)
{
double d1,d2,d3;
d1=Point_calculateDistance(P1,P2);
d2=Point_calculateDistance(P2,P3);
d3=Point_calculateDistance(P3,P1);
if(pow(d1,2)+pow(d2,2)==pow(d3,2)|| pow(d2,2)+pow(d3,2)==pow(d1,2)|| pow(d3,2)+pow(d1,2)==pow(d2,2))
{
    return 1;
}
else return 0;
}
```

## Question 33

```
1 // You can print the values to stdout for debugging
2 int isRightTriangle(Point *P1, Point *P2, Point *P3)
3 {
4     // Write your code here
5 }
6
7
```

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **countOccurrence** return an integer representing the count of occurrences of given value in the input list.

The function/method **countOccurrence** accepts three arguments - *len*, an integer representing the size of the input list, *value*, an integer representing the given value and *arr*, a list of integers, representing the input list.

The function/method **countOccurrence** compiles successfully but fails to return the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 int countOccurrence( int len, int value, int *arr)
3 {
4     int i=0, count = 0;
5     while(i<len){
6         if(arr[i]==value)
7             count += 1;
8     }
9     return count;
10 }
11 }
```

## Solution :

```
int countOccurrence(int len,int value,int *arr)
{
int i=0,count=0;
while(i<len)
{
if(arr[i]==value)
count+=1;
return value;
}
}
```

## Question 34

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method `drawPrintPattern` accepts `num`, an integer.

The function/method `drawPrintPattern` prints the first `num` lines of the pattern shown below.

For example, if `num` = 3, the pattern should be:

```
11
1111
11111
```

The function/method `drawPrintPattern` compiles successfully but fails to get the desired result for some test cases due to incorrect implementation of the function/method. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void drawPrintPattern(int num)
3 {
4     int i,j,print = 1;
5     for(i=1;i<=num;i++)
6     {
7         for(j=1;j<=2*i;j++)
8         {
9             printf("%d ",print);
10        }
11        printf("\n");
12    }
13 }
14 }
```

## Solution:

```
void drawPrintPattern(int num)
{
    for(i=1;i<=num;i++)
    {
        for(j=1;j<=2*i;j++)
        {
            printf("%d ",print);
        }
        printf("\n");
    }
}
```

## Question 35

You are required to fix all the logical error in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We do not expect you to modify the approach.

The function/method `sameElementCount` returns an integer representing the number of elements of the input list which

are even numbers and equal to the element to its right. For example, if the input list is

[4 4 4 1 8 4 1 1 2 2] then the

function/method should return the output '3' as it has three similar groups i.e,

(4, 4), (4, 4), (2, 2).

The function/method sameElementCount accepts two arguments - size, an integer representing the size of the input list and inputList, a list of integers representing the input list.

The function/method compiles successfully but fails to return the desired result for some test cases due to incorrect implementation of the function/method sameElementCount. Your task is to fix the code so that it passes all the test cases.

Note:

In a list, an element at index i is considered to be on the left of index i+1 and to the right of index i-1.

The last element of the input list does not have any element next to it which makes it incapable to satisfy the second condition and hence should not be counted.

**Problem:**

```
int sameElementCount(int size,int *inputList)
{
    int i,count=0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
            count++;
    }
    return count;
}
```

**Solution:**

```
int sameElementCount(int size,int *inputList)
{
    int i,count=0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
            count++;
    }
    return count;
}
```

**Question 36**

You are required to complete the given code. You can click on Compile & Run anytime to check the compilation/execution \ status of the program. You can use printf() to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the main() function as it is not required.

Code Approach: For this question, you will need to complete the code as in given

implementation. We do not expect you to modify the approach. The function/method median accepts two arguments - size and inputList, an integer representing the length of a list and a list of integers, respectively.

The function/method median is supposed to calculate and return an integer representing the median of elements in the input list. However, the function/method median works only for odd-length lists because of incomplete code.

You must complete the code to make it work for even-length lists as well. A couple of other functions/methods are available, which you are supposed to use inside the function/method median to complete the code. Helper Description

The following function is used to represent a quick\_select and is already implemented in the default code (Do not write this definition again in your code): int quick\_select(int inputList, int start\_index, int end\_index, int median\_order)

```
/*It calculate the median value This can be called as -  
quick_select(inputList, start_index, end_index, median_order) where  
median order is the half length of the inputlist
```

**Problem:**

```
float median(int size,int *inputList)
```

```
{  
    int start_index=0;  
    int end_index=size-1;  
    float res=-1;  
    if(size%2!=0)  
    {  
        int median_order=((size+1)/2);  
  
        res=(float)quick_select(inputList,start_index,end_index,median-  
order);  
    }  
    else  
    {  
        //enter your code  
    }  
    return res;  
}
```

**Solution:**

```

float median(int size,int *inputList)
{
    int start_index=0;
    int end_index=size-1;
    float res=-1;
    if(size%2!=0)
    {
        int median_order=((size+1)/2);

        res=(float)quick_select(inputList,start_index,end_index,median-
order);
    }
    else
    {
        int median_order=((size+1)/2);

        res=((float)(quick_select(inputList,start_index,end_index,median-
order)+

                quick_select(inputList,start_index,end_index,median-
order))/2);
    }
    return res;
}

```

**Testcase 1:**

Input: 5,  
[2, 40, 23, 52, 37]

Expected Return Value:  
37.00

**Testcase 2:**

Input: 6,  
[-24, -16, -8, -4, -54, -1]

Expected **Return** Value:

-12.00

### Question 37

given three point find whether they forms a right angled triangle or not.

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
}
```

find the distance between three points. let them be a,b,c.

if  $a^2+b^2=c^2$

then it will be right triangle

**Solution:**

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
    double d1,d2,d3,max,min,mid;
    d1=point_calculateDistance(p1,p2);
    d2=point_calculateDistance(p2,p3);
    d3=point_calculateDistance(p1,p3);
    max = (d1>d2)?(d1>d3?d1:d3):(d2>d3)?d2:d3;
    min = (d1<d2)?(d1<d3?d1:d3):(d2<d3)?d2:d3;
    mid = (d1+d2+d3)-(min+max);
    if(max*max==min*min+mid*mid)
        return 1;
    else
        return 0;
}
```

### Question 38

given three point find whether they forms a right angled triangle or not.

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
}
```

find the distance between three points. let them be a,b,c.

if  $a^2+b^2=c^2$

then it will be right triangle

**Solution:**

```
int isRightTriangle(Point *p1,Point *p2,Point *p3)
{
    double d1,d2,d3,max,min,mid;
    d1=point_calculateDistance(p1,p2);
    d2=point_calculateDistance(p2,p3);
    d3=point_calculateDistance(p1,p3);
    max = (d1>d2)?(d1>d3?d1:d3):(d2>d3)?d2:d3;
    min = (d1<d2)?(d1<d2?d1:d3):(d2<d3)?d2:d3;
    mid = (d1+d2+d3)-(min+max);
    if(max*max==min*min+mid*mid)
        return 1;
    else
        return 0;
}
```

## Question 39

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method `drawPrintPattern` accepts `num`, an integer. The function/method `drawPrintPattern` prints the first `num` lines of the pattern shown below.

For example, if `num = 3`, the pattern should be:

```
11
1111
111111
```

The function/method `drawPrintPattern` compiles successfully but fails to get the desired result for some test cases due to incorrect implementation of the function/method. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void drawPrintPattern(int num)
3 {
4     int i,j,print = 1;
5     for(i=1;i<=num;i++)
6     {
7         for(j=1;j<=2*i;j++)
8         {
9             printf("%d ",print);
10        }
11        printf("\n");
12    }
13 }
14 }
```

## Code

```
void drawPrintPattern(int num)
{
    int i,j,print=1;
```

```

for(i=1;i<=num;i++)
{
    for(j=1;j<=2*i;j++)
    {
        printf("%d ",print);
    }
    printf("\n");
}

```

## Question 40

You are required to fix all the logical error in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required.

**Code Approach:** For this question, you will need to complete the code as in given implementation. We **do not** expect you to modify the approach.

The function/method *sameElementCount* returns an integer representing the number of elements of the input list which are even numbers and equal to the element to its right. For example, if the input list is [4 4 4 18 4 11 2 2] then the function/method should return the output '3' as it has three similar groups i.e, (4, 4), (4, 4), (2, 2).

The function/method *sameElementCount* accepts two arguments - *size*, an integer representing the size of the input list and *inputList*, a list of integers representing the input list.

The function/method compiles successfully but fails to return the desired result for some test cases due to incorrect implementation of the function/method *sameElementCount*. Your task is to fix the code so that it passes all the test cases.

**Note:**

In a list, an element at index *i* is considered to be on the left of index *i+1* and to the right of index *i-1*. The last element of the input list does not have any element next to it which makes it incapable to satisfy the second condition and hence should not be counted.

```

1 // You can print the values to stdout for debugging
2 int sameElementCount(int size, int *inputList)
3 {
4     int i,count =0;
5     for(i=0;i<size-1;i++)
6     {
7         if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
8             count++;
9     }
10    return count;
11 }
12
13

```

## Code

```

int sameElementCount(int size,int *inputList)
{
    int i,count=0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
            count++;
    }
    return count;
}

```

## Question 41

You are required to fix all the logical error in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the `main()` function as it is not required.

**Code Approach:** For this question, you will need to complete the code as is given implementation. We **do not** expect you to modify the approach.

The function/method `sameElementCount` returns an integer representing the number of elements of the input list which are even numbers and equal to the element to its right. For example, if the input list is [4 4 4 18 4 11 2 2] then the function/method should return the output 3 as it has three similar groups i.e. [4, 4], [4, 4], [2, 2].

The function/method `sameElementCount` accepts two arguments - `size`, an integer representing the size of the input list and `inputList`, a list of integers representing the input list.

The function/method compiles successfully but fails to return the desired result for some test cases due to incorrect implementation of the function/method `sameElementCount`. Your task is to fix the code so that it passes all the test cases.

**Note:**

In a list, an element at index  $i$  is considered to be on the left of index  $i+1$  and to the right of index  $i-1$ . The last element of the input list does not have any element next to it which makes it incapable to satisfy the second condition and hence should not be counted.

```
1 // You can print the values to stdout for debugging
2 int sameElementCount(int size, int *inputList)
3 {
4     int i, count = 0;
5     for(i=0;i<size-1;i++)
6     {
7         if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
8             count++;
9     }
10    return count;
11 }
12
13 }
```

**Testcase 1:**

**Input:**

11,

[1, 5, 5, 2, 2, 7, 8, 6, 6, 9, 18]

**Expected Return Value:**

2

**Testcase 2:**

**Input:**

5,

[13, 12, 12, 13, 14]

**Expected Return Value:**

1

```
1 // You can print the values to stdout for debugging
2 int sameElementCount(int size, int *inputList)
3 {
4     int i, count = 0;
5     for(i=0;i<size-1;i++)
6     {
7         if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
8             count++;
9     }
10    return count;
11 }
12
13 }
```

## Solution

```
int sameElementCount(int size, int *inputList)
{
    int i, count=0;
    for(i=0;i<size-1;i++)
    {
        if((inputList[i]%2==0)&&(inputList[i]==inputList[i+1]))
            count++;
    }
    return count;
}
```

## Question 42

You are required to fix all logical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use `printf()` to debug your code. The submitted code should be logically/syntactically correct and pass all testcases.

**Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method **manchester** print space-separated integers with the following property: for each element in the input list `arr`, if the bit `arr[i]` is the same as `arr[i-1]`, then the element of the output list is 0. If they are different, then its 1. For the first bit in the input list, assume its previous bit to be 0. This encoding is stored in a new list.

The function/method **manchester** accepts two arguments - `len`, an integer representing the length of the list and `arr` and `arr`, a list of integers, respectively. Each element of `arr` represents a bit - 0 or 1.

For example - if `arr` is [0 1 0 0 1 1 0], the function/method should print an list [0 1 1 0 1 0 0 1].

The function/method compiles successfully but fails to print the desired result for some test cases due to logical errors. Your task is to fix the code so that it passes all the test cases.

```
1 // You can print the values to stdout for debugging
2 void manchester(int len, int* arr)
3 {
4     int* res = (int*)malloc(sizeof(int)*len);
5     res[0] = arr[0];
6     for(int i = 1; i < len; i++){
7         res[i] = (arr[i]-arr[i-1]);
8     }
9     for(int i = 0; i < len; i++){
10        printf("%d ",res[i]);
11    }
}
```

**Testcase 1:**

**Input:**  
0, [1, 1, 0, 0, 1, 1, 0]

**Expected Return Value:**

1 0 1 0 1 1

**Testcase 2:**

**Input:**  
0, [0, 0, 0, 1, 0, 1, 1, 1]

**Expected Return Value:**

0 0 0 1 1 1 0 0

```
1 // You can print the values to stdout for debugging
2 void manchester(int len, int* arr)
3 {
4     int* res = (int*)malloc(sizeof(int)*len);
5     res[0] = arr[0];
6     for(int i = 1; i < len; i++){
7         res[i] = (arr[i]-arr[i-1]);
8     }
9     for(int i = 0; i < len; i++){
10        printf("%d ",res[i]);
11    }
}
```

## Solution

```
void manchester(int len, int* arr)
{
    int* res = (int*)malloc(sizeof(int)*len);
    res[0] = arr[0];
    for(int i = 1; i < len; i++)
    {
        res[i] = !(arr[i] == arr[i-1]);
    }
    for(int i = 0; i < len; i++)
    {
        printf("%d ", res[i]);
    }
}
```

## Question 43

You are required to fix all syntactical errors in the given code. You can click on *Compile & Run* anytime to check the compilation/execution status of the program. You can use *printf()* to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the *main()* function as it is not required. **Code Approach:** For this question, you will need to correct the given implementation. We **do not** expect you to modify the approach or incorporate any additional library methods.

The function/method ***multiplyNumber*** returns an integer representing the multiplicative product of the maximum two of three input numbers. The function/method ***multiplyNumber*** accepts three integers- *numA*, *numB* and *numC*, representing the input numbers.

The function/method ***multiplyNumber*** compiles unsuccessfully due to syntactical error. Your task is to debug the code so that it passes all the test cases.

**Testcase 1:**

**Input:**  
5, 7, 4

**Expected Return Value:**

35

**Testcase 2:**

**Input:**  
11, 12, 13

**Expected Return Value:**

156

```
1 // You can print the values to stdout for debugging
2 int multiplyNumber(int numA, int numB, int numC)
3 {
4     int result,min,max,mid;
5     max=(numA>numB)?(numA>numC)?numA:numC):((numB>numC)?numB:numC);
6     min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
7     mid=(numA+numB+numC)-(min+max);
8     result=(max*int mid);
9     return result;
10 }
```

```
1 // You can print the values to stdout for debugging
2 int multiplyNumber(int numA, int numB, int numC)
3 {
4     int result,min,max,mid;
5     max=(numA>numB)?(numA>numC)?numA:numC):((numB>numC)?numB:numC);
6     min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
7     mid=(numA+numB+numC)-(min+max);
8     result=(max*int mid);
9     return result;
10 }
```

## Solution:

```
int multiplyNumber(int numA, int numB, int numC)
{
    int result,min,max,mid;
    max=(numA>numB)?((numA>numC)?numA:numC):((numB>numC)?numB:numC);
    min=(numA<numB)?((numA<numC)?numA:numC):((numB<numC)?numB:numC);
    mid=(numA+numB+numC)-(min+max);
    result=(max*mid);
    return result;
}
```

